



Aerospace Vehicle Control And Perception Systems

A Model-Based Approach

version 0.0.1

Jordan D. Larson

©Jordan D. Larson 2024
ALL RIGHTS RESERVED

This PDF is made available for personal use only.
Commercial reproduction is prohibited, except as authorized by the author.

Contents

| | |
|---|-----------|
| Contents | 1 |
| 0 Preface and Acknowledgments | 7 |
| 0.1 Preface: Model-Based Design | 7 |
| 0.2 Acknowledgments | 10 |
| I Dynamical Systems and Linear Control Theory | 11 |
| 1 Introductory Dynamical Systems Theory | 12 |
| 1.1 Introduction to Dynamical Systems | 12 |
| 1.2 Equilibrium and Linearization of Time-Invariant Systems | 19 |
| 1.3 General Solution and Modal Analysis of LTI Systems | 24 |
| 1.4 Free, Impulse, and Step Responses of SISO LTI Systems | 30 |
| 1.5 Sinusoidal and Frequency Responses of SISO LTI Systems | 39 |
| 1.6 Frequency Responses of Fundamental SISO LTI Systems | 44 |
| 2 Continuous-Time SISO LTI Feedback Control Theory | 54 |
| 2.1 Introduction to Control Systems | 54 |
| 2.2 Classical Feedback Control System Analysis | 60 |
| 2.3 Classical Feedback Control System Stability Margins | 64 |
| 2.4 SISO Loop-Shaping Control Design Requirements | 69 |
| 2.5 Fundamental SISO Loop-Shaping Control Stages | 76 |
| 2.6 SAS and PID Control Design | 83 |
| 2.7 Classical Feedback Control System Performance Constraints | 90 |
| 3 Continuous-Time MIMO Linear Feedback Control Theory | 98 |

| | | |
|-----------|---|------------|
| 3.1 | MIMO LTI System Analysis | 98 |
| 3.2 | MIMO LTI Feedback Control Systems | 108 |
| 3.3 | Robust Stability of LTI Systems | 117 |
| 3.4 | Command Tracking Control System Design | 123 |
| 3.5 | LTV Systems Theory and Control | 126 |
| 4 | Continuous-Time Linear Optimal Control Theory | 132 |
| 4.1 | Introduction to Optimal Control | 132 |
| 4.2 | Introductory Energy-, Time-, and Fuel-Optimal Control | 138 |
| 4.3 | Semidefinite Programming in Optimal Control | 145 |
| 4.4 | Introductory \mathcal{H}_2 Optimal Control | 149 |
| 4.5 | Introductory \mathcal{H}_{∞} Optimal Control | 155 |
| 4.6 | Robust Optimal Control of LPV Systems | 164 |
| 5 | Discrete-Time Systems and Linear Control Theory | 168 |
| 5.1 | Discrete-Time Linear Systems | 168 |
| 5.2 | Discrete-Time Linear Feedback Control Systems | 175 |
| 5.3 | Discrete-Time Linear-Quadratic Regulator | 181 |
| 5.4 | Iterative and Extended Linear-Quadratic Regulator | 185 |
| 5.5 | Receding-Horizon Linear-Quadratic Regulator | 191 |
| 6 | Linear-in-Control Systems and Adaptive Control Theory | 195 |
| 6.1 | Linear-in-Control Systems and Dynamic Inversion | 195 |
| 6.2 | Gain-Scheduled Adaptive Control | 200 |
| 6.3 | Model Reference Adaptive Control | 200 |
| 6.4 | L_1 Adaptive Control | 205 |
| 6.5 | Concurrent Learning Adaptive Control | 205 |
| II | Aerospace Vehicle Dynamics and Control Systems | 206 |
| 7 | Aerospace Vehicle Attitude and Steady-Flight Models | 208 |
| 7.1 | Introduction to Aerospace Vehicles | 208 |
| 7.2 | Reference Frames and Rotations | 222 |
| 7.3 | Aerospace Vehicle Reference Frames and Euler Angles | 231 |
| 7.4 | Steady-Flight Conditions for Aerospace Vehicles | 240 |
| 8 | Introductory Rigid Aerospace Vehicle Dynamics | 250 |
| 8.1 | Introductory Rigid Aerospace Vehicle Dynamics | 250 |
| 8.2 | Introductory Rigid Airplane Trim Analysis | 257 |
| 8.3 | Introductory Linearized Rigid Airplane Dynamics | 263 |
| 8.4 | Airplane Stability and Flying Qualities | 269 |
| 8.5 | Rigid Rotary-Wing Aircraft Dynamics | 275 |
| 8.6 | Axisymmetric Spacecraft Attitude Dynamics and Stability | 278 |

| | |
|--|------------|
| 8.7 Gravity-Gradient Satellite Attitude Dynamics and Stability | 283 |
| 9 Advanced Rigid Aerospace Vehicle Dynamics | 290 |
| 9.1 Advanced Airplane Trim Analysis and Linearized Dynamics | 290 |
| 9.2 Advanced Mass Effects on Aerospace Vehicle Dynamics | 298 |
| 9.3 Atmospheric Effects on Aerospace Vehicle Dynamics | 303 |
| 9.4 Earth Frame Effects on Aerospace Vehicle Dynamics | 309 |
| 9.5 Relative Orbital Satellite Dynamics | 318 |
| 10 Elastic Aerospace Vehicle Dynamics | 321 |
| 10.1 Elastic-Body Dynamics | 321 |
| 10.2 Elastic Vehicle Mean-Axes | 336 |
| 10.3 Elastic Aerospace Vehicle Dynamics | 342 |
| 10.4 Elastic Airplane Dynamics | 349 |
| 10.5 Elastic Satellite Dynamics | 357 |
| 11 Introductory Aerospace Vehicle Control Systems | 358 |
| 11.1 Aerospace Actuation Systems | 358 |
| 11.2 Aerospace Guidance and Control Systems | 363 |
| 11.3 Introductory Airplane Attitude Control Systems | 369 |
| 11.4 Spacecraft Dual-Spin Stabilization Systems | 379 |
| 11.5 Precision Approach Guidance Systems | 382 |
| 12 Advanced Aerospace Vehicle Control Systems | 388 |
| 12.1 Aerospace Structural Mode Control Systems | 388 |
| 12.2 Airplane Total Energy and Heading Control Systems | 388 |
| 12.3 Rotary-Wing Attitude Control Systems | 393 |
| 12.4 Spacecraft Four-Wheel Attitude Control Systems | 393 |
| 12.5 Spacecraft Attitude-Maneuver Systems | 393 |
| 12.6 Planar Intercept Guidance Systems | 394 |
| 12.7 L_1 Guidance Systems | 400 |
| 12.8 Powered Explicit Guidance Systems | 400 |
| III Probability and Estimation Theory for Perception | 401 |
| 13 Probability Theory for Perception | 402 |
| 13.1 Probability Theory and Random Variables | 402 |
| 13.2 Random Vectors | 412 |
| 13.3 Fundamental Probability Distributions for Perception | 420 |
| 13.4 Random Processes | 427 |
| 13.5 Random Dynamical Systems for Control and Perception | 434 |
| 13.6 Detection and Hypothesis Testing Theory | 441 |
| 13.7 Random Finite Sets | 446 |

| | |
|---|------------|
| <i>CONTENTS</i> | 4 |
| 14 Optimal Parameter Estimation Theory | 453 |
| 14.1 Introduction to Optimal Parameter Estimation | 453 |
| 14.2 Introductory Least-Squares Regression | 460 |
| 14.3 Advanced Batch Least-Squares Methods | 467 |
| 14.4 Introductory Bayesian Optimal Parameter Estimation | 472 |
| 14.5 Advanced Bayesian Optimal Parameter Estimation | 479 |
| 15 Optimal Linear State Estimation Theory | 488 |
| 15.1 Introduction to Optimal State Estimation | 488 |
| 15.2 Discrete-Time Kalman Filtering and Smoothing | 495 |
| 15.3 Alternative Discrete-Time Kalman Filtering and Smoothing | 502 |
| 15.4 Generalized Noise for Discrete-Time Kalman Filtering | 511 |
| 15.5 Discretization and Continuous-Time Kalman Filtering | 518 |
| 15.6 Multiple Model Filtering | 524 |
| 16 Introductory Nonlinear Bayesian State Estimation Theory | 532 |
| 16.1 Introduction to Nonlinear Bayesian State Estimation | 532 |
| 16.2 Alternative Extended Kalman Filtering and Smoothing | 541 |
| 16.3 Statistical Linearization and State Estimation | 548 |
| 16.4 Sigma-Point Kalman Filtering and Smoothing | 556 |
| 16.5 Particle Filtering | 563 |
| 16.6 Alternative Particle Filtering and Smoothing | 569 |
| 17 Advanced Kalman Filtering Theory | 577 |
| 17.1 Error-State Kalman Filtering | 577 |
| 17.2 Multi-Modal and Heavy-Tailed Kalman Filtering | 580 |
| 17.3 Variational Bayes Kalman Filtering | 580 |
| 17.4 Distributed Kalman Filtering | 581 |
| IV Aerospace Vehicle Perception Systems | 582 |
| 18 Aerospace Sensor Systems | 583 |
| 18.1 Inertial Sensors | 583 |
| 18.2 Radionavigation Systems | 588 |
| 18.3 Radar, Electro-Optical, and Infrared Sensors | 599 |
| 18.4 Air Data Systems | 603 |
| 18.5 Fuel and Battery Monitoring Systems | 605 |
| 18.6 Fault and Integrity Monitoring Systems | 609 |
| 18.7 Clocks and Timing Systems | 615 |
| 19 Attitude Determination Systems | 620 |
| 19.1 Introduction to Attitude Determination Systems | 620 |
| 19.2 Gyroscopic Attitude Determination Systems | 624 |

| | |
|--|------------|
| <i>CONTENTS</i> | 5 |
| 19.3 Single-Frame Attitude Determination Systems | 625 |
| 19.4 Attitude and Heading Reference Systems | 629 |
| 20 Positioning Systems | 633 |
| 20.1 Introduction to Positioning Systems | 633 |
| 20.2 Pseudorange Multilateration | 644 |
| 20.3 Phase-Based Positioning | 651 |
| 20.4 GNSS-Based Positioning | 656 |
| 21 Introductory Navigation Systems | 662 |
| 21.1 Introduction to Navigation Systems | 662 |
| 21.2 Inertial Navigation Systems | 666 |
| 21.3 GNSS-Aided Navigation Systems | 672 |
| 21.4 Baro- and Terrain-Aided Navigation Systems | 677 |
| 21.5 Feature Map-Aided Navigation Systems | 677 |
| 22 Introductory Target Tracking Systems | 678 |
| 22.1 Introduction to Target Tracking Systems | 678 |
| 22.2 Probabilistic Data Association Filtering | 687 |
| 22.3 Multi-Hypothesis Tracking | 694 |
| 22.4 Uncoupled-Coordinate Motion Models | 699 |
| 22.5 Turning Motion Models | 704 |
| 22.6 Ballistic Flight Motion Models | 710 |
| 23 Advanced Navigation & Target Tracking Systems | 720 |
| 23.1 Multi-Object Filtering Based on Random Finite Sets | 720 |
| 23.2 Probability Hypothesis Density Filtering | 723 |
| 23.3 Cardinalized Probability Hypothesis Density Filtering | 727 |
| 23.4 Multi-Bernoulli Filtering | 732 |
| 23.5 Labeled Multi-Bernoulli Filtering | 733 |
| 23.6 Extended Object Tracking | 735 |
| 23.7 Simultaneous Localization and Mapping | 735 |
| V Appendices | 736 |
| A Fundamental Mathematical Concepts | 737 |
| A.1 Concepts from Set Theory | 737 |
| A.2 Concepts from Linear Algebra | 739 |
| A.3 Concepts from Complex Analysis and Stability Criterion | 748 |
| B Airplane Stability and Control Derivative Models | 754 |
| B.1 Finite-Wing Theory and Component Build-Up Model | 754 |
| B.2 Longitudinal Stability and Control Derivatives | 761 |

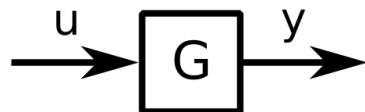
| | |
|---|------------|
| <i>CONTENTS</i> | 6 |
| B.3 Lateral-Directional Stability and Control Derivatives | 770 |
| Bibliography | 777 |
| Index | 778 |

Preface and Acknowledgments

0.1 Preface: Model-Based Design

Model-based design provides a common design environment facilitating communication, data analysis, design verification and validation (V&V) between various development groups for the entire system. Because modern computers allow the use of complex structures and extensive software code, engineers typically use model-based design with advanced simulation tools to provide rapid prototyping of new systems, including aerospace systems. In this way, engineers can locate and correct errors early in system design, when the impact of design modifications are minimized. Furthermore, design reuse for upgraded or expanded systems is made easier. However, as model-based design uses a coverall approach to standard embedded systems development, the time it takes to port between modeling software and embedded systems can outweigh the temporal value for alternative lab-based design.

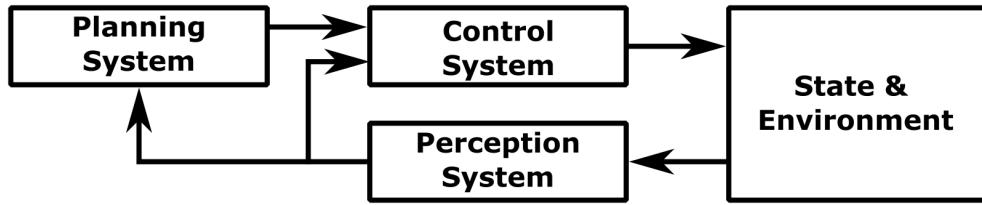
Model-based design software typically uses graphical modeling tools to provide a very generic and unified graphical modeling environment by reducing the complexity of model designs by breaking them into hierarchies of individual design blocks. This helps design engineers to conceptualize the entire system, typically with a **graphical user interface (GUI)**. The connection between signals and systems in these GUIs are typically depicted using **block diagrams**, for which each “block” represents a system and each “arrow” represents a signal. An example of a basic block diagram can be drawn as



which consists of a system G with a single input signal u and single output signal y .

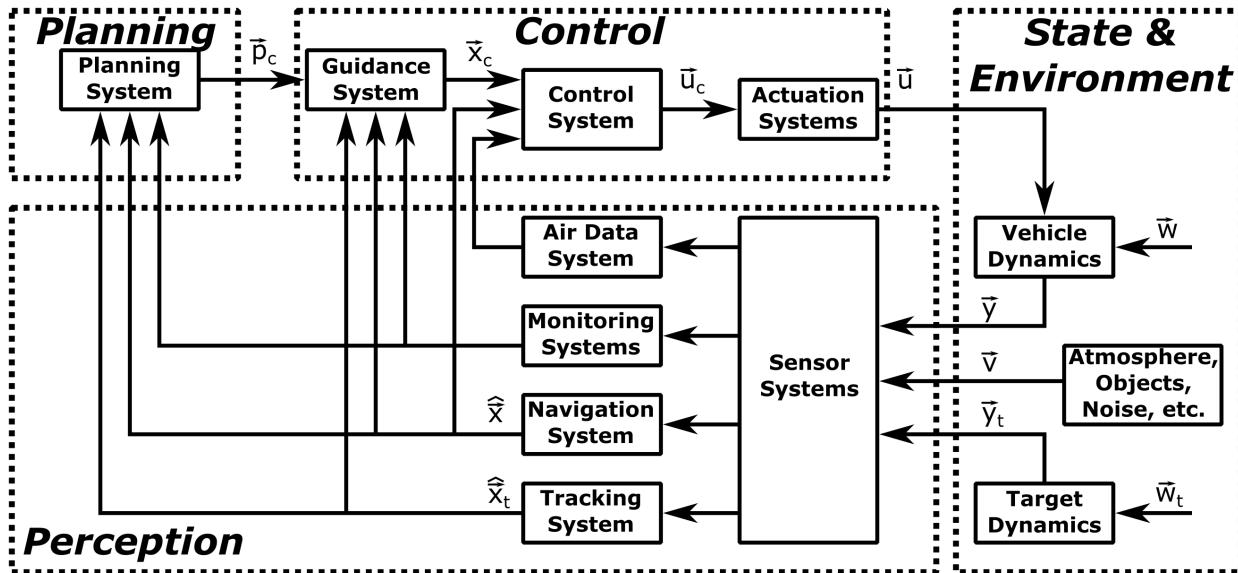
Aerospace Planning, Control, and Perception Systems

In modern aerospace applications, the design of the **planning, control, and perception** systems is fundamental to the operations of aerospace vehicles, also known as flight vehicles. These three systems can be understood as connected together through the following



where the planning system determines what path or trajectory the aerospace vehicle should travel (or fly) to reach its current target, the control system executes the path or trajectory for the aerospace vehicle by actuating the control inputs while the perception system(s) senses and estimates the state of the aerospace vehicle, the state of its target(s), and the state of the environment. Thus, one can see that for aerospace vehicles, perception systems provide updates to the planning system for the flight path as well as a (guidance and) control system to follow the planned trajectory.

Furthermore, one can represent the operation of an aerospace vehicle as an aerospace **planning, control, and perception** system in the following block diagram.



where the control block contains both the guidance, control, and actuation systems while the perception block contains the air data, monitoring, navigation, and target tracking systems. Of these systems, the majority of the content in this textbook is focused on trajectory guidance systems, attitude control systems, navigation

systems, and target tracking systems. However, the model-based theoretical principles contained in this textbook offer the reader the ability to design a much wider array of aerospace systems, including the air data systems, battery management systems, and fault detection systems which are discussed in lesser detail.

This textbook approaches the design of components of control and perception systems for aerospace vehicles from a model-based theoretic primarily relying on linear, time-invariant systems theory for deterministic, uncertain, and stochastic systems with extensions provided for linear, parameter-varying and nonlinear systems. Thus, the models inherent to this approach rely heavily on flight dynamics models which borrows from the field of dynamical systems theory. Each part of this textbook corresponds to content that is covered in an entire course at the senior undergraduate or graduate level.

While planning systems for aerospace vehicles are mentioned throughout this textbook, the general design principles of these systems is not a part of the scope of this textbook as the path/trajectory planning algorithms and mission planning concepts of operation (CONOPS) are often context-specific and vary dramatically on the type of and application for the aerospace vehicle. However, it is important to remember that the use of trajectory/path planning algorithms is integral and vital to the operation of aerospace vehicles especially in air and space traffic command, control, and communications.

It would be beneficial for a first-time reader of this book to be familiar with the following mathematical concepts from an undergraduate course(s) in linear algebra and differential equations.

- Vector and matrix equation manipulation in three-dimensions
- Eigenvalue decomposition
- Partial fraction decomposition
- (Inverse) Laplace transform
- Solutions to first- and second-order ordinary differential equations (ODEs)

Other mathematical concepts necessary for this textbook are provided in Appendix A. Lastly, it should be noted that in this textbook, \forall denotes “for all” and $:$ denotes “such that.”

Model-Based Design of Control and Perception Systems

During the 1920s, control and perception systems were first implemented in electric powerplant industries. These large process facilities used process controllers for regulating continuous variables, e.g. temperature, pressure, and flow rate, with electrical relays. Throughout the 1930s and 1940s, control and perception systems began to see use in the automotive and aviation industries. In the 1950s and 1960s, the space race generated interest in embedded control and perception systems. Here, engineers constructed control and perception systems that could be part of the end product, e.g. engine control units and flight simulators. In the 1970s, computer-based control and perception systems were introduced and became standard bringing about a drastic shift in control and perception system design and has led to the modern use of model-based design for control and perception systems.

Model-based design of control and perception systems can be described as four steps. Step one is **plant modeling** which consists of identifying the system model to be controlled, i.e. the **plant** which comes from the “powerplant” in early days of control and perception systems. Plant modeling can be based on first principles models or machine learning. First principles modeling implements a physics-derived

mathematical model for the plant dynamics, e.g. Newton-Euler EOMs. Machine learning processes raw data from a real-world system and uses learning algorithms to identify the data-driven model for the plant dynamics. Step two is **control and perception design** which consists of **control and perception analysis** and **control and perception synthesis**. First, the mathematical plant model developed in the first step is used to identify suitable **design requirements** for the controlled plant. Based on these identified requirements, a control strategy and a perception strategy are chosen and then a control system and a perception system are synthesized producing an explicit **control law** and an **observer law**, i.e. two mathematical algorithms for computing the plant inputs automatically based on external commands and sensor data.

Step three is **system simulation** which may be performed as multiple simulations with lower/higher fidelity plant models as well as offline/real-time simulations. These simulations allow specification, requirements, and modeling errors to be found immediately, rather than later in the overall system design effort. Real-time simulation can be done by testing the control system, the perception system, and plant on a real-time modeling computer, also known as **software-in-the-loop (SIL)** simulation. Sometimes the control system, perception system may also be implemented on the embedded system hardware and simulated with a real-time plant and sensor model, also known as **hardware-in-the-loop (HIL)** simulation. Step four is **validation and verification (V&V)** of the control and perception system. Here, the control and perception systems are implemented on the embedded system hardware and tested with the actual plant, e.g. aerospace vehicle, and sensors. As the control and perception systems are highly unlikely to work exactly the same on actual plant and sensors as in simulation, an iterative debugging process is carried out by analyzing the test results and updating the control system and perception system designs until the design requirements are met in actuality.

0.2 Acknowledgments

The author acknowledges his former students who assisted in proofreading this textbook.

- Aabhash Bhandari
- Eric Becker
- Adam Hallmark
- Bo Landess
- Tuan Luong
- Holden McNerney
- Isaiah Newell
- Ryan Thomas
- Michael Van Oteghem
- Vaughn Weirens

Part I

Dynamical Systems and Linear Control Theory

Introductory Dynamical Systems Theory

1.1 Introduction to Dynamical Systems

The study of **flight dynamics and control (FDC)** most generally falls under the theory of signals and systems. A **signal** is a mathematical description of how a parameter varies with time. Signals can be continuous or discrete in time, as well as continuous or discrete in the values the parameter may take. **Analog signals** are continuous-time and continuous-valued. **Digital signals** are discrete-time and discrete-valued. **Discrete-time signals** are discrete in time and continuous-valued. Signals that are continuous-time and discrete-valued rarely occur and do not have a particular name. A **system** is any process that produces output signals in response to input signals. The output signal from a system is also known as the **system response**. A system may be characterized by its different properties.

A fundamental characterization is by how the output signal or system response depends on time. A **static system** produces an output signal that is not time-dependent while a **dynamical system** produces an output signal as a time-dependent quantity that evolves according to a fixed mathematical rule, also known as the **dynamics equation**. Here the term “dynamical” is used to correspond to the mathematical abstraction which is modeling some real-world dynamic process. For many physical dynamical systems, e.g. flight vehicles, the dynamics equation is also known as the **equation of motion (EOM)**. The study of dynamical systems encompasses three broad topics: simulation, system identification (SID), and control. Each of these concepts will be introduced in this introductory part of the textbook for FDC.

A dynamical system may be characterized by its signal types. **Analog systems** have only analog signals while **digital systems** have only digital signals. Importantly, any electronic dynamical system contains both analog and digital signals at some level which are traditionally analyzed separately along with the analog-to-digital and/or digital-to-analog converters.

A dynamical system may be characterized by the form of its dynamics equation. A **linear system** satisfies the **principle of superposition** which states that when one adds multiple input signals together as well as multiplies them by arbitrary scalars, then the overall system output signal will be equivalent to the added and scaled output signals of the individual input signals. A **nonlinear system** does not satisfy the principle of superposition. A **time-invariant system** has a dynamics equation that does not depend *explicitly* on time. A

time-varying system has a dynamics equation that does depend *explicitly* on time. Lastly, a **deterministic system** will always produce the same output signal for a given input signal. If this is not true, it is a **stochastic system**.

Lastly, a dynamical system may be characterized by its number of input and output signals. If a system has no inputs, it is an **autonomous system**. Otherwise, it may be a **single input, single output (SISO) system**; a **single input, multiple outputs (SIMO) system**; a **multiple inputs, single output (MISO) system**; or a **multiple inputs, multiple outputs (MIMO) system**.

This chapter of the textbook introduces the mathematical theory for continuous-time, **linear time-invariant (LTI)**, deterministic dynamical systems. Unfortunately, no dynamical system is perfectly linear, time-invariant, or deterministic. First, linearity implies that the operation of a system can be scaled to arbitrarily large magnitudes which is not physically possible in the extreme. Second, time-invariance is violated by aging effects that change the outputs of systems over time. Lastly, thermal noise and other random phenomena ensure that the operation of any system will have some degree of random or stochastic behavior. However, despite this limitation, these assumptions greatly simplify the mathematical theory of dynamical systems while still providing valuable insight and intuition into FDC.

Continuous-Time System Models

The simplest mathematical model for representing continuous-time dynamical systems explicitly uses time derivatives in a system's dynamics equation. Continuous-time SISO systems can be modeled using univariate **ordinary differential equations (ODEs)** which have the general form

$$\frac{d^n y}{dt^n} = f \left(t, y, \frac{dy}{dt}, \dots, \frac{d^{n-1}y}{dt^{n-1}}, u, \frac{du}{dt}, \dots, \frac{d^m u}{dt^m} \right) \quad (1.1)$$

where $y(t)$ is the single **output signal**, $u(t)$ is the single **input signal**, and n is the **order of the ODE** with $p \leq n$ denoting a **proper ordinary differential equation**. Furthermore, the ODE is said to be **unforced** if $f()$ is not a function of an input u , i.e.

$$\frac{d^n y}{dt^n} = f \left(t, y, \frac{dy}{dt}, \dots, \frac{d^{n-1}y}{dt^{n-1}} \right) \quad (1.2)$$

The solution of the ODE, i.e. the specific output $y(t)$ due to some specific input $u(t)$, requires the use of a **boundary condition** for the system output at some time t where a boundary condition given at $t = 0$ is also known as an **initial condition**. It should be noted that from here on, this textbook will use Newton's dot notation to represent time derivatives up to the third order and a bracketed exponent for higher orders, i.e.

$$y^{[n]}(t) = f \left(t, y, \dot{y}, \dots, y^{[n-1]}, u, \dot{u}, \dots, u^{[m]} \right) \quad (1.3)$$

A multivariate extension of ODEs for continuous-time MIMO systems is the **continuous-time state-space model** which can be defined as the general form of two equations

$$\begin{aligned} \dot{\vec{x}}(t) &= f(t, \vec{x}, \vec{u}) \\ \vec{y}(t) &= h(t, \vec{x}, \vec{u}) \end{aligned} \quad (1.4)$$

where $f : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$, $h : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_y} \rightarrow \mathbb{R}^{n_u}$, $\vec{u}(t) \in \mathbb{R}^{n_u}$ is the **input vector** of n_u input signals, $\vec{y}(t) \in \mathbb{R}^{n_y}$ is the **output vector** of n_y output signals, and $\vec{x} \in \mathbb{R}^{n_x}$ is the **state vector** of the n_x^{th} -order dynamical system. The first vector-valued differential equation is the **dynamics equation**, also known as the **state equation**, for continuous-time state-space systems. The second vector-valued algebraic equation is the **output equation** and relates the output vector to the state and input vectors. In this representation, the state of the system \vec{x} is dynamically *controlled* by the input \vec{u} and is *observed* through the output \vec{y} with the function.

It should be noted that this vector-valued state-space system can be rewritten as n_x first order ODEs for the state equation and n_y algebraic equations for the output in the general form

$$\begin{aligned}\dot{x}_1 &= f_1(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ &\vdots = \vdots \\ \dot{x}_{n_x} &= f_{n_x}(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ y_1 &= h_1(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ &\vdots = \vdots \\ y_{n_y} &= h_{n_y}(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u})\end{aligned}\tag{1.5}$$

where x_i denotes the i^{th} element of \vec{x} , u_j denotes the j^{th} element of \vec{u} , and y_k denotes the k^{th} element of \vec{y} .

The study of general continuous-time systems is an advanced topic and is discussed in subsequent parts of this textbook. This introductory chapter on dynamical systems focuses on the analysis of linear, time-invariant (LTI) systems. As stated previously, linear systems satisfy the **principle of superposition** described by two properties: scaling and additivity. Nonlinear systems do not, in general, satisfy these properties. To demonstrate these properties, let the output $y_1(t)$ be a solution to a linear ODE with input $u_1(t)$ and zero initial conditions and let $y_2(t)$ be the solution with $u_2(t)$ and zero initial conditions. The scaling property states that for any real number c , the solution of the linear ODE with input $u_s(t) = cu_1(t)$ and zero initial conditions is given by $y_s(t) = cy_1(t)$. The additivity property states that the solution of the linear ODE with input $u_a(t) = u_1(t) + u_2(t)$ and zero initial conditions is $y_a(t) = y_1(t) + y_2(t)$.

With these properties in mind, one can show that **linear ODEs** have the general form

$$y^{[n]}(t) + a_{n-1}(t)y^{[n-1]}(t) + \dots + a_1(t)\dot{y}(t) + a_0(t)y(t) = b_m(t)u^{[m]}(t) + \dots + b_1(t)\dot{u}(t) + b_0(t)u(t)\tag{1.6}$$

and **continuous-time linear state-space representations** have the general form

$$\begin{aligned}\dot{\vec{x}}(t) &= A(t)\vec{x}(t) + B(t)\vec{u}(t) \\ \vec{y}(t) &= C(t)\vec{x}(t) + D(t)\vec{u}(t)\end{aligned}\tag{1.7}$$

where $A(t) \in \mathbb{R}^{n_x \times n_x}$ matrix is the **state matrix**. The $B(t) \in \mathbb{R}^{n_x \times n_u}$ matrix is the **input matrix**. The $C(t) \in \mathbb{R}^{n_y \times n_x}$ matrix is the **output matrix** and $D(t) \in \mathbb{R}^{n_y \times n_u}$ is the **feedthrough matrix**.

As stated previously, time-invariant systems have dynamics equations that do not depend explicitly on time. Thus, **time-invariant ODEs** have the general form

$$y^{[n]}(t) = f(y, \dot{y}, \dots, y^{[n-1]}, u, \dot{u}, \dots, u^{[m]})\tag{1.8}$$

and **continuous-time time-invariant state-space representations** have the general form

$$\begin{aligned}\dot{\vec{x}}(t) &= f(\vec{x}, \vec{u}) \\ \vec{y}(t) &= h(\vec{x}, \vec{u})\end{aligned}\tag{1.9}$$

Continuous-Time Linear Time-Invariant System Models

Thus, **linear time-invariant ordinary differential equations (LTI ODEs)** have the general form

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_m u^{[m]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t)\tag{1.10}$$

and **continuous-time linear time-invariant state-space models** have the general form

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned}\tag{1.11}$$

Thus, note that LTI systems theory borrows heavily from mathematical methods in differential equations and linear algebra which will be utilized throughout this textbook.

With this general form in mind, note that a particular LTI state-space model can be denoted by the quadruple (A, B, C, D) . It should also be noted that there are different choices for (A, B, C, D) for modeling the same dynamical system in terms of the input-to-output relationship, $\vec{u}(t)$ -to- $\vec{y}(t)$, although the internal state, $\vec{x}(t)$, will be different for each model. However, if one wishes to set $\vec{y} = \vec{x}$, then one may set the output matrix to the **identity matrix**, i.e.

$$C = I = \begin{bmatrix} 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \vdots & 1 & 0 \\ 0 & 0 & \vdots & 0 & 1 \end{bmatrix}\tag{1.12}$$

and the feedthrough matrix to the **zero matrix**, i.e.

$$D = \begin{bmatrix} 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix}\tag{1.13}$$

For proper n^{th} order LTI ODE, i.e. $m \leq n$, one can rewrite this as

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_n u^{[n]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t)\tag{1.14}$$

which has an infinite number of equivalent state-space models. One common state-space model conversion is the **Controllable Canonical Form (CCF)** which is performed as follows. Let

$$\begin{aligned}x_1 &= y \\ x_2 &= \dot{y} \\ &\vdots = \vdots \\ x_{n_x-1} &= y^{[n-1]} \\ x_{n_x} &= y^{[n]}\end{aligned}\tag{1.15}$$

Then, the matrices of the CCF state-space system can be defined as

$$A = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ -a_0 & -a_1 & -a_2 & \cdots & -a_{n-1} \end{bmatrix} \quad (1.16)$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \quad (1.17)$$

$$C = [(b_0 - a_0 b_n) \quad (b_1 - a_1 b_n) \quad \cdots \quad (b_{n-2} - a_{n-2} b_n) \quad (b_{n-1} - a_{n-1} b_n)] \quad (1.18)$$

$$D = b_n \quad (1.19)$$

Note that if $b_n = 0$ then the formula is much simpler. With this conversion in mind, this textbook will primarily use the state-space model instead of the ODE model in its presentation of dynamical aerospace systems which are intrinsically multivariate and when they are not, one can always easily convert from an ODE to state-space.

Another important, albeit abstracted, SISO LTI system model is the **transfer function** which uses a change of variables from the real variable t to a complex variable s under zero initial conditions by the **Laplace transform**

$$f(s) = \mathcal{L}\{f(t)\} = \int_0^\infty f(t)e^{-st} dt \quad (1.20)$$

and vice versa by the **inverse Laplace transform**

$$f(t) = \mathcal{L}^{-1}\{f(s)\} = \frac{1}{2\pi j} \lim_{T \rightarrow \infty} \int_{\gamma-jT}^{\gamma+jT} f(s)e^{st} ds \quad (1.21)$$

where j here represents $\sqrt{-1}$, a dynamical systems notation resulting from electrical systems already using lowercase i for current. From this transform, one can define the conversion of derivatives as

$$x^{[i]}(t) = s^i x(s) - \sum_{k=1}^i s^{i-k} x^{[k-1]}(0) \quad (1.22)$$

and linear expressions as

$$ax_1(t) + bx_2 = aX_1(s) + bX_2(s) \quad (1.23)$$

For a LTI ODE

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_m u^{[m]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t) \quad (1.24)$$

to find the transfer function, $G(s)$, which “transfers” the transformed input $u(s)$ to the transformed output $y(s)$ through simple multiplication, i.e.

$$y(s) = G(s)u(s) \quad (1.25)$$

one can convert to the equivalent transfer function model using the Laplace domain with zero initial conditions as

$$s^n y(s) + a_{n-1} s^{n-1} y(s) + \cdots + a_1 s y(s) + a_0 y(s) = b_m s^m u(s) + \cdots + b_1 s u(s) + b_0 u(s) \quad (1.26)$$

Rearranging, one has

$$\left(s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0 \right) y(s) = (b_m s^m + \cdots + b_1 s + b_0) u(s) \quad (1.27)$$

or

$$y(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} u(s) \quad (1.28)$$

Thus, by definition of the transfer function, the **standard transfer function form** can be defined as

$$G(s) = \frac{b_m s^m + \cdots + b_1 s + b_0}{s^n + a_{n-1} s^{n-1} + \cdots + a_1 s + a_0} \quad (1.29)$$

Notably, the numerator and denominator of a transfer function is a polynomial, thus, each has real and/or complex roots. A **zero**, z , of a transfer function is a root of the numerator while a **pole**, p , of a transfer function is a root of the denominator. These terms derive from the behavior of the transfer function’s magnitude, $|G(s)|$, in the complex plane, namely, $|G(s)| \rightarrow 0$ as $s \rightarrow z$ and $|G(s)| \rightarrow \infty$ as $s \rightarrow p$. Plotting $|G(s)|$ over the complex plane, this infinite value will look like a tent pole in a three dimensional plot. With this in mind, it should be noted that one can always factor a transfer function into

$$G(s) = \frac{K(s - z_1) \cdots (s - z_m)}{(s - p_1) \cdots (s - p_n)} \quad (1.30)$$

A transfer function has a **pole-zero cancellation** if any poles and zeros have the same value. If there are no pole-zero cancellations, then a transfer function is said to be a **minimal realization**. One can also factor a transfer function into a **partial-fraction decomposition**, i.e.

$$G(s) = \frac{r_1}{(s - p_1)} + \cdots + \frac{r_n}{(s - p_n)} \quad (1.31)$$

where r_i is the i^{th} constant factor known as a **residue** which will be zero if a pole-zero cancellation exists for p_i . An important result that one can use with the transfer function model for LTI systems is the **Final Value Theorem (FVT)** which states if every pole of a transfer function $F(s)$ is not purely imaginary *except* $F(s)$ has, at most, a single pole at the origin. Then,

$$\lim_{t \rightarrow \infty} f(t) = \lim_{s \rightarrow 0} s F(s) \quad (1.32)$$

where $s \rightarrow 0$ denotes s approaching through the positive numbers.

For reference, some additional conversions for functions in the t and s domains are provided in the following table which are often used for converting between $u(t)$ and $u(s)$ and $y(t)$ and $y(s)$, respectively.

| Function | t Domain ($\forall t \geq 0$) | s Domain |
|---|-----------------------------------|-------------------------------------|
| Unit Step | 1 | $\frac{1}{s}$ |
| Exponential | e^{-at} | $\frac{1}{s+a}$ |
| Time power $n \in \mathbb{N} +$ Exponential | $\frac{t^{n-1}}{(n-1)!} e^{-at}$ | $\frac{1}{(s+a)^n}$ |
| Sine | $\sin \omega t$ | $\frac{\omega}{s^2 + \omega^2}$ |
| Cosine | $\cos \omega t$ | $\frac{s}{s^2 + \omega^2}$ |
| Exponentially Decaying Sine | $e^{-at} \sin \omega t$ | $\frac{\omega}{(s+a)^2 + \omega^2}$ |
| Exponentially Decaying Cosine | $e^{-at} \cos \omega t$ | $\frac{s+a}{(s+a)^2 + \omega^2}$ |

For a continuous-time MIMO LTI systems as a state-space model, i.e.

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned}\tag{1.33}$$

one transforms to the Laplace domain with zero initial conditions as

$$\begin{aligned}s\vec{x}(s) &= A\vec{x}(s) + B\vec{u}(s) \\ \vec{y}(s) &= C\vec{x}(s) + D\vec{u}(s)\end{aligned}\tag{1.34}$$

To find the **transfer function matrix**, $[G(s)]$, which maps each input to each output through a matrix of transfer functions, i.e.

$$\vec{y}(s) = [G(s)]\vec{u}(s)\tag{1.35}$$

one requires finding $\vec{x}(s)$ in terms of $\vec{u}(s)$ and substituting into the output equation. Thus,

$$s\vec{x}(s) - A\vec{x}(s) = B\vec{u}(s)\tag{1.36}$$

$$(sI - A)\vec{x}(s) = B\vec{u}(s)\tag{1.37}$$

$$\vec{x}(s) = (sI - A)^{-1}B\vec{u}(s)\tag{1.38}$$

where $(sI - A)^{-1}$ is known as the **resolvent** of A .

Then, by substitution

$$\vec{y}(s) = C(sI - A)^{-1}B\vec{u}(s) + D\vec{u}(s)\tag{1.39}$$

$$\vec{y}(s) = \left(C(sI - A)^{-1}B + D\right)\vec{u}(s)\tag{1.40}$$

Thus, by definition of the transfer function matrix, **standard transfer function matrix** can be defined as

$$[G(s)] = C(sI - A)^{-1}B + D\tag{1.41}$$

which is a $n \times m$ matrix of transfer function elements, $G_{ij}(s)$, i.e.

$$[G(s)] = \begin{bmatrix} G_{11}(s) & \cdots & G_{1m}(s) \\ \vdots & \ddots & \vdots \\ G_{n1} & \cdots & G_{nm}(s) \end{bmatrix}\tag{1.42}$$

Lastly, one sometimes may use the **polynomial-matrix model** defined as

$$P(s)\vec{y}(s) = Q(s)\vec{u}(s) \quad (1.43)$$

where the system transfer function matrix is given by

$$[G(s)] = P^{-1}(s)Q(s) \quad (1.44)$$

References

For more information, please refer to the following

- Schmidt, D. K., “10.1 Linear System Analysis - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.2 State-Space Models,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 144-154
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.3 Transfer Function Models,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 155-170

1.2 Equilibrium and Linearization of Time-Invariant Systems

Equilibrium and Trim of Dynamical Systems

For ODEs, an output-input pair (\bar{y}, \bar{u}) is an **equilibrium point** if all derivatives of y and u are zero for all $t > 0$, i.e. if

$$f(t, \bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) = 0 \quad (1.45)$$

is a valid solution for all $t \geq 0$. Thus, if one initializes $y(t) = \bar{y}$ at $t = 0$ and sets $u(t) = \bar{u}$ for $t \geq 0$, then $y(t) = \bar{y}$ for all $t \geq 0$, i.e. y is “steady.” Note that one may choose two variables \bar{y} and \bar{u} to solve for univariate equilibrium equation which may lead to multiple, even infinite, solutions. Thus, equilibrium points for systems with inputs are also known as a **trim point**, as one is said to be “trimming” the system input \bar{u} to accomplish equilibrium at a particular trim output, \bar{y} . By extension, for state-space models, a state-input pair (\vec{x}, \vec{u}) is an **equilibrium point** if \vec{x} is zero for all $t > 0$, i.e. if

$$f(t, \vec{x}, \vec{u}) = 0 \quad (1.46)$$

is a valid solution for all $t \geq 0$. Similarly, if one initializes $\vec{x}(t) = \vec{x}$ at $t = 0$ and sets $\vec{u}(t) = \vec{u}$ for $t \geq 0$, then $\vec{x}(t) = \vec{x}$ and $\vec{y} = h(\vec{x}, \vec{u})$ for all $t \geq 0$, i.e. \vec{x} is “steady.” Here, there also may be multiple, even infinite, solutions for a particular **trim state**, \vec{x} , since there are fewer equations than free variables for the vector-valued state equation, i.e. n_x elements and $n_x + n_u$ free variables.

An important characterization of system equilibrium points is their stability which was first studied by Lyapunov in his dissertation *The General Problem of Stability of Motion*. With the condition $u(t) = \bar{u}$, an

equilibrium point \bar{y} of an ODE is defined as **stable in the sense of Lyapunov (SISL)** if for some $|\epsilon| > 0$ with $y(0) = \bar{y} + \epsilon$, there exists some δ for which $|y(t)| < |\bar{y} + \delta|$ as $t \rightarrow \infty$, i.e. if one initializes the dynamical system “near enough” to equilibrium, the system will “remain close” to the equilibrium point. Moreover, if a δ exists for all $\epsilon \in \mathbb{R}$, then the equilibrium point is defined as **globally stable**. If $y \rightarrow \bar{y}$ as $t \rightarrow \infty$, then the equilibrium point is defined as **asymptotically stable**. Lastly, if $y \rightarrow \bar{y}$ for all $\epsilon \in \mathbb{R}$, then the equilibrium point is defined as **globally asymptotically stable (GAS)**.

To qualitatively assess the stability of equilibrium points, Lyapunov developed two methods. The first method of Lyapunov states that if the *linearized dynamical system* about the equilibrium point is strictly stable, then the original dynamical system is also stable for some “stability neighborhood” about the equilibrium point. The size of this “stability neighborhood” is directly related to the effects of the neglected higher-order terms (HOT) in the linearization. Thus, for highly nonlinear ODEs one typically analyzes the system’s stability using Lyapunov’s second method as the linearized stability neighborhood is too small to be practically useful. However, this part of the textbook focuses on LTI FDC and will only consider Lyapunov’s first method for time-invariant systems and will provide methods for assessing a flight vehicle’s stability and designing control systems. A later part of this textbook will discuss Lyapunov’s second method.

Linearization of Time-Invariant ODEs

The linearization of time-invariant ODEs requires the **Taylor series** of a univariate function $f(y)$ about \bar{y} which is defined as

$$f(y) = f(\bar{y}) + \left[\frac{df}{dy} \right]_{\bar{y}} (y - \bar{y}) + \frac{1}{2} \left[\frac{d^2 f}{dy^2} \right]_{\bar{y}} (y - \bar{y})^2 + \dots \quad (1.47)$$

which is an infinite series of increasing order. The **linearization of $f(y)$ about \bar{y}** approximates this series by the inclusion of the zeroth and first order terms, i.e. the constant and proportional terms, which produces a linear function as

$$f(y) \approx f(\bar{y}) + \left[\frac{df}{dy} \right]_{\bar{y}} (y - \bar{y}) \quad (1.48)$$

which can be said to approximate the true function. Thus, the linearization is also known as a **first-order approximation**. The difference between the exact solution and the linearization is known as the **linearization error** and is a result of the removal of **higher-order terms (HOT)** from the Taylor series.

In addition, it should also be noted that often one uses the substitution

$$y = \bar{y} + \Delta y \quad (1.49)$$

where Δy is a **perturbation** about the point \bar{y} . This type of substitution is known as **perturbation form**. Then, the linearization can be rewritten as

$$f(\bar{y} + \Delta y) \approx f(\bar{y}) + \left[\frac{df}{dy} \right]_{\bar{y}} \Delta y \quad (1.50)$$

which is known as the fundamental **linearization theorem** in **small-perturbation theory**, also known as **small-disturbance theory**.

The general form for a time-invariant ODE is given by

$$y^{[n]} = f \left(y, \dot{y}, \dots, y^{[n-1]}, u, \dot{u}, \dots, u^{[m-1]} \right) \quad (1.51)$$

Then, modeling the output signal in perturbation form as

$$y(t) = \bar{y} + \Delta y(t) \quad (1.52)$$

and the input signal in perturbation form as

$$u(t) = \bar{u} + \Delta u(t) \quad (1.53)$$

the linearization of a time-invariant ODE about a trim point, (\bar{u}, \bar{y}) , can be written as

$$\Delta y^{[n]} + a_{n-1}\Delta y^{[n-1]} + \cdots + a_1\Delta \dot{y} + a_0\Delta y = b_m\Delta u^{[m]} + \cdots + b_1\Delta \dot{u} + b_0\Delta u \quad (1.54)$$

where the coefficients of this LTI ODE are

$$a_0 = -\frac{\partial f}{\partial y}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.55)$$

$$a_i = -\frac{\partial f}{\partial y^{[i]}}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.56)$$

for $i = 1, \dots, n-1$, and

$$b_0 = \frac{\partial f}{\partial u}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.57)$$

$$b_j = \frac{\partial f}{\partial u^{[j]}}(\bar{y}, 0, \dots, 0, \bar{u}, 0, \dots, 0) \quad (1.58)$$

for $j = 1, \dots, m$. Note that the negative signs appear for the a coefficients due to the standard form for LTI ODEs where the left side of the equation contains all y terms and the right side contains all u terms. Lastly, note that the derivatives of $y(t)$ and $u(t)$ are equivalent to those of $\Delta y(t)$ and $\Delta u(t)$ since \bar{y} and \bar{u} are constants.

Linearization of Time-Invariant State-Space Models

The linearization of time-invariant state-space models requires the **Taylor Series** for multivariate function $f(\vec{x}, \vec{u})$ about the vector pair $(\bar{\vec{x}}, \bar{\vec{u}})$ is

$$\dot{\vec{x}}(t) = f(\vec{x}, \vec{u}) = f(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] (\vec{x} - \bar{\vec{x}}) + \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] (\vec{u} - \bar{\vec{u}}) + \text{HOT} \quad (1.59)$$

or

$$\vec{y}(t) = h(\vec{x}, \vec{u}) = h(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] (\vec{x} - \bar{\vec{x}}) + \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] (\vec{u} - \bar{\vec{u}}) + \text{HOT} \quad (1.60)$$

where $\left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right]$ is the **Jacobian** of $f()$. Thus, multivariate linearization is also known as **Jacobian linearization**.

Defining the state, input, and output **perturbation vectors** about constants $\bar{\vec{x}}$, $\bar{\vec{u}}$, and $\bar{\vec{y}}$ as

$$\Delta \vec{x}(t) = \vec{x}(t) - \bar{\vec{x}} \quad (1.61)$$

$$\Delta \vec{u}(t) = \vec{u}(t) - \bar{\vec{u}} \quad (1.62)$$

$$\Delta \vec{y}(t) = \vec{y}(t) - \bar{\vec{y}} \quad (1.63)$$

and recognizing for trim, $f(\bar{\vec{x}}, \bar{\vec{u}}) = 0$ and $h(\bar{\vec{x}}, \bar{\vec{u}}) = \bar{\vec{y}}$, one has

$$\Delta \dot{\vec{x}}(t) = \dot{\vec{x}}(t) = f(\vec{x}, \vec{u}) = f(\bar{\vec{x}}, \bar{\vec{u}}) + \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta \vec{x}(t) + \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta \vec{u}(t) + \text{HOT} \quad (1.64)$$

$$\Delta \vec{y}(t) = \vec{y}(t) - \bar{\vec{y}} = h(\vec{x}, \vec{u}) - h(\bar{\vec{x}}, \bar{\vec{u}}) = \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta \vec{x}(t) + \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta \vec{u}(t) + \text{HOT} \quad (1.65)$$

Recall the general form for a time-invariant state-space model

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}, \vec{u}) \\ \vec{y} &= h(\vec{x}, \vec{u}) \end{aligned} \quad (1.66)$$

Next, setting

$$A = \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_1}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_{n_x}}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \quad (1.67)$$

$$B = \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_1}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_{n_x}}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \quad (1.68)$$

$$C = \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] = \begin{bmatrix} \frac{\partial h_1}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_1}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_{n_y}}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \quad (1.69)$$

$$D = \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] = \begin{bmatrix} \frac{\partial h_1}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_1}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_{n_y}}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \quad (1.70)$$

yields an approximate LTI state-space model about $\bar{\vec{x}}$ and $\bar{\vec{u}}$ as

$$\begin{aligned} \Delta \dot{\vec{x}}(t) &\approx A \Delta \vec{x}(t) + B \Delta \vec{u}(t) \\ \Delta \vec{y}(t) &\approx C \Delta \vec{x}(t) + D \Delta \vec{u}(t) \end{aligned} \quad (1.71)$$

Linearizations of Trigonometric Functions

As an aside, an important linearization is for the trigonometric sine and cosine functions about any angle $\bar{\theta}$. It can be shown that the Taylor series for sine and cosine are

$$\sin(\bar{\theta}) = \sin(\bar{\theta}) + \cos(\bar{\theta})(\theta - \bar{\theta}) - \frac{1}{2!} \sin(\bar{\theta})(\theta - \bar{\theta})^2 - \frac{1}{3!} \cos(\bar{\theta})(\theta - \bar{\theta})^3 + \dots \quad (1.72)$$

and

$$\cos(\bar{\theta}) = \cos(\bar{\theta}) - \sin(\bar{\theta})(\theta - \bar{\theta}) - \frac{1}{2!} \cos(\bar{\theta})(\theta - \bar{\theta})^2 + \frac{1}{3!} \sin(\bar{\theta})(\theta - \bar{\theta})^3 + \dots \quad (1.73)$$

where θ must be in radians. Thus, the linearizations for sine and cosine about $\bar{\theta}$ are given by

$$\sin(\bar{\theta}) \approx \sin \bar{\theta} + \cos(\bar{\theta})(\theta - \bar{\theta}) \quad (1.74)$$

and

$$\cos(\bar{\theta}) \approx \cos \bar{\theta} - \sin(\bar{\theta})(\theta - \bar{\theta}) \quad (1.75)$$

Of particular note, for $\bar{\theta} = 0$ rad, the linearization simplifies to

$$\sin \theta \approx \theta \quad (1.76)$$

and

$$\cos \theta \approx 1 \quad (1.77)$$

In this case, one can also approximate

$$\tan \theta \approx \theta \quad (1.78)$$

These three approximations are commonly called the **small angle approximations** and will be used throughout this textbook. These are decent approximations when $\theta < 15^\circ = 0.2618$ rad, producing a linearization error of 0.0028, 0.034, 0.0061 and for sine, cosine, and tangent, respectively.

References

For more information, please refer to the following

- Schmidt, D. K., “1.1 Small Perturbation Theory for Nonlinear Systems,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 1-2
- Schmidt, D. K., “1.4 Vector Differentiation,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 10-14
- Schmidt, D. K., “1.6 Small Perturbation Analysis Revisited,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 18-21
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.7 Numerical Linearization,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 199-205

1.3 General Solution and Modal Analysis of LTI Systems

General Solution to LTI Systems

First, consider a proper first-order LTI ODE, i.e.

$$\dot{y}(t) + a_0 y(t) = b_0 u(t) \quad (1.79)$$

with boundary conditions, $y(t_0)$. From calculus, the general solution can be shown to be

$$y(t) = y_H(t) + y_P(t) \quad (1.80)$$

where $y_H(t)$ is the **ODE homogeneous solution**, i.e. the solution for $u(t) = 0$, and $y_P(t)$ is the **ODE particular solution** due to $u(t) \neq 0$.

Consider the **exponential function** of $a(t - t_0)$, i.e.

$$e^{a(t-t_0)} = \sum_{i=0}^{\infty} \frac{1}{i!} [a(t-t_0)]^i \quad (1.81)$$

with derivative

$$\frac{d}{dt} e^{a(t-t_0)} = \sum_{i=0}^{\infty} \frac{1}{i!} ai + 1(t-t_0)^i = ae^{a(t-t_0)} \quad (1.82)$$

Thus, the homogeneous solution to the first-order ODE is simply

$$y(t) = e^{-a_0(t-t_0)} y(t_0) \quad (1.83)$$

and the particular solution can be computed using a integrating factor $e^{a_0 t}$

$$e^{a_0 t} \dot{y}_P(t) + e^{a_0 t} a_0 y_P(t) = e^{a_0 t} b_0 u(t) \quad (1.84)$$

By recognizing the product rule of the left side, one has

$$\frac{d}{dt} (e^{a_0 t} y_P(t)) = e^{a_0 t} b_0 u(t) \quad (1.85)$$

Next, integrating over the input signal from t_0 to t , one has

$$e^{a_0 t} y_P(t) = \int_{t_0}^t e^{a_0 \tau} b_0 u(\tau) d\tau \quad (1.86)$$

Finally, multiplying both sides by $e^{-a_0 t}$ yields

$$y_P(t) = \int_{t_0}^t e^{-a_0(t-\tau)} b_0 u(\tau) d\tau \quad (1.87)$$

Thus, the general solution for the first-order system is

$$y(t) = e^{-a_0(t-t_0)} y(t_0) + \int_{t_0}^t e^{-a_0(t-\tau)} b_0 u(\tau) d\tau \quad (1.88)$$

It is important to note that the fundamental characteristic of this first-order system depends on the value of a_0 . Namely, if $a_0 > 0$, then the exponential terms in the general solution will decay to 0 as $t \rightarrow \infty$, i.e. the system output will reach some bounded output due to the integrated input signal. Thus, the LTI system is stable.

Next, consider the general LTI state-space form

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) \\ \vec{y}(t) &= C\vec{x}(t) + D\vec{u}(t)\end{aligned}\tag{1.89}$$

with boundary conditions $\vec{x}(t_0)$. By analogy, the general solution for the vector-valued first-order state equation can be shown to be

$$\begin{aligned}\vec{x}(t) &= e^{A(t-t_0)}\vec{x}(t_0) + \int_{t_0}^t e^{A(t-\tau)}B\vec{u}(\tau)d\tau \\ \vec{y}(t) &= C\left[e^{A(t-t_0)}\vec{x}(t_0) + \int_{t_0}^t e^{A(t-\tau)}B\vec{u}(\tau)d\tau\right] + D\vec{u}(t)\end{aligned}\tag{1.90}$$

where

$$e^{A(t-t_0)} = \sum_{k=0}^{\infty} \frac{1}{k!} [A(t-t_0)]^k = \lim_{k \rightarrow \infty} (I + \frac{1}{k}A(t-t_0))^k\tag{1.91}$$

is known as the **matrix exponential function** of $A(t-t_0)$, also known as the **LTI state-transition matrix**, $\Phi(t, t_0)$. Also by analogy to a_0 , analysis of the values of the state matrix, A , provides the stability analysis of the general LTI system.

Characteristic Polynomial of LTI Systems

Consider a n^{th} -order LTI system represented by the standard transfer function model, i.e.

$$y(s) = G(s)u(s)\tag{1.92}$$

where, using the standard ODE conversion, one has

$$G(s) = \frac{b_ms^m + \cdots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0}\tag{1.93}$$

and the **ODE characteristic equation** can be defined as

$$\lambda^n + a_{n-1}\lambda^{n-1} + \cdots + a_1\lambda + a_0 = 0\tag{1.94}$$

where the left side of this equation is called the **ODE characteristic polynomial**. The solutions to this equation are the n roots of the ODE characteristic polynomial and may be repeated and will either be real numbers or complex-conjugate pairs.

This can be generalized for all LTI systems by the standard transfer function matrix model, i.e.

$$\vec{y}(s) = [G(s)]\vec{u}(s)\tag{1.95}$$

where, using the standard state-space conversion, one has

$$[G(s)] = C(sI - A)^{-1}B + D \quad (1.96)$$

Furthermore, by use of the **adjugate**, also known as the **classical adjoint** of A instead of the matrix inverse, i.e.

$$[G(s)] = C \left(\frac{\text{adj}(sI - A)}{\det[sI - A]} \right) B + D \quad (1.97)$$

one can see that each transfer function element, $G_{ij}(s)$ of $[G(s)]$, will have the same denominator polynomial given by $\det[sI - A]$, and is known as the **state-space characteristic polynomial**.

Thus, the **state-space characteristic equation** for general LTI systems can be defined as

$$\det[\lambda I - A] = 0 \quad (1.98)$$

The solutions to this equation are the n_x roots of the state-space characteristic polynomial and may be repeated and will either be real numbers or complex-conjugate pairs. Lastly, the roots of the characteristic polynomials are also known as the **system poles**. Notably, if there are no pole-zero cancellations, then the poles of any $G_{ij}(s)$ are also the system poles. Furthermore, each distinct real pole and each distinct complex-conjugate pair is called a **system mode** which are important in analyzing the characteristics of LTI systems.

Modal Representation of LTI Systems

To analyze the modes of an LTI system, first, note that the LTI system poles are equivalent to the eigenvalues of A , λ_i . These eigenvalues solve the **eigenvalue equations** for right eigenvectors, \vec{v}_i , i.e.

$$A\vec{v}_i = \lambda_i\vec{v}_i \quad (1.99)$$

and for left eigenvectors, $\vec{\mu}_i^T$, i.e.

$$\vec{\mu}_i^T A = \lambda_i \vec{\mu}_i^T \quad (1.100)$$

If there are n_x distinct eigenvalues, then A is **diagonalizable** and one can perform an **eigenvalue decomposition** of A as

$$A = V\Lambda V^{-1} \quad (1.101)$$

where Λ is the diagonal matrix of n_x eigenvalues, i.e.

$$\Lambda = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_{n_x} \end{bmatrix} \quad (1.102)$$

and V is the matrix of n_x linearly independent right eigenvectors, i.e.

$$V = [\vec{v}_1 \ \cdots \ \vec{v}_{n_x}] \quad (1.103)$$

and V^{-1} is the matrix of n_x linearly independent left eigenvectors, i.e.

$$V^{-1} = \begin{bmatrix} \vec{\mu}_1^T \\ \vdots \\ \vec{\mu}_{n_x}^T \end{bmatrix} \quad (1.104)$$

Next, define the **modal state vector**, \vec{z} , such that

$$\vec{x} = V\vec{z} = \vec{v}_1 z_1(t) + \cdots + \vec{v}_{n_x} z_{n_x}(t) \quad (1.105)$$

and

$$\vec{z} = V^{-1}\vec{x} = \begin{bmatrix} \vec{\mu}_1^T \vec{x} \\ \vdots \\ \vec{\mu}_{n_x}^T \vec{x} \end{bmatrix} \quad (1.106)$$

where notably, $z_{n_x}(t)$, is dimensionless. Then, substituting into the state equation, one has

$$V\dot{\vec{z}}(t) = AV\vec{z}(t) + B\vec{u}(t) \quad (1.107)$$

or

$$\dot{\vec{z}}(t) = V^{-1}AV\vec{z}(t) + V^{-1}B\vec{u}(t) \quad (1.108)$$

Thus, one can form the **Jordan canonical form (JCF)** of the state-space system as

$$\begin{aligned} \dot{\vec{z}}(t) &= \Lambda\vec{z}(t) + V^{-1}B\vec{u}(t) \\ \vec{y}(t) &= CV\vec{z}(t) + D\vec{u}(t) \end{aligned} \quad (1.109)$$

where the state-transition matrix for diagonalizable A is now

$$e^{\Lambda(t-t_0)} = \begin{bmatrix} e^{\lambda_1(t-t_0)} & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & e^{\lambda_{n_x}(t-t_0)} \end{bmatrix} \quad (1.110)$$

and this model explicitly **decouples** the state equation into n_x LTI ODEs, i.e.

$$\dot{z}_i(t) = \lambda_i z_i(t) + \vec{\mu}_i^T B\vec{u}(t), \quad i = 1, \dots, n_x \quad (1.111)$$

Lastly, recalling **Euler's formula**, i.e.

$$e^{j\omega t} = \cos(\omega t) + j\sin(\omega t) \quad (1.112)$$

note that the exponential of a complex-conjugate pair $\sigma \pm j\omega$ for an eigenvalue can be rewritten as

$$e^{(\sigma \pm j\omega)(t-t_0)} = e^{\sigma(t-t_0)} (\cos(\omega(t-t_0)) \pm j\sin(\omega(t-t_0))) \quad (1.113)$$

Modal Analysis of LTI Systems

Consider the zero-input response where the n_x modal states are given by

$$z_i(t) = e^{\lambda_i(t-t_0)} z_i(t_0) = e^{\lambda_i(t-t_0)} \vec{\mu}_i^T \vec{x}(t_0) \quad (1.114)$$

and the output response is given by

$$\vec{y}(t) = C (\vec{v}_1 z_1(t) + \cdots + \vec{v}_{n_x} z_{n_x}(t)) \quad (1.115)$$

Thus, one can see the modal states are characterized by the eigenvalues and left eigenvectors while the output response is additionally characterized by the right eigenvectors and the modal states. As these eigenvalues will occur as real numbers or in complex-conjugate pairs, the modal states may be complex-valued which describes the oscillatory nature of the modal responses. Regardless, the real part of each individual mode allows one to characterize the stability of each mode as well as the LTI system as a whole. In particular, a mode is **stable** if the real part of $\lambda_i < 0$, **marginally stable** if the real part of $\lambda_i = 0$, and **unstable** if the real part of $\lambda_i > 0$. Furthermore, the LTI system is **stable** if *all* the modes are strictly stable, the LTI system is **marginally stable** if *all* the modes are stable or marginally stable, and if *any* mode is unstable, the LTI system is **unstable**. As these eigenvalues may be complex-valued, one can also state that all the system poles located in the left half of the complex plane, i.e. the **left half plane (LHP)**, correspond to stable modes.

Furthermore, each mode corresponds to a unique **modal time constant** which describes its exponential decay/growth rate, i.e.

$$\tau = -\frac{1}{\text{Real}(\lambda_i)} \quad (1.116)$$

where $\text{Real}(\lambda)$ represents the real part of λ . Thus, if $\tau > 0$, the mode is stable. This also demonstrates modes have faster decay rates on the output for modes with higher eigenvalues. For complex-conjugate pairs, $\lambda = -\frac{1}{\tau} \pm j\omega_d$, one also defines the **modal damped frequency**, ω_d , which are related to the **modal undamped natural frequency**, ω_n , and **modal damping ratio**, $0 < \zeta_i < 1$, by

$$\omega_d = \omega_n \sqrt{1 - \zeta^2} \quad (1.117)$$

and

$$\tau = \frac{1}{\zeta \omega_n} \quad (1.118)$$

and one can compare

$$(\lambda + \frac{1}{\tau} + j\omega_d)(\lambda + \frac{1}{\tau} - j\omega_d) \quad (1.119)$$

to the standard second-order form

$$\lambda^2 + 2\zeta\omega_n\lambda + \omega_n^2 \quad (1.120)$$

and note $\lambda = -\zeta\omega_n \pm j\omega_n\sqrt{1 - \zeta^2}$.

Consider the transfer function matrix model using the eigenvalue decomposition with $D = 0$, i.e.

$$[G(s)] = C(sI - M\Lambda M^{-1})^{-1}B \quad (1.121)$$

which one can show results in the matrix

$$[G(s)] = CM \begin{bmatrix} \frac{1}{s-\lambda_1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \frac{1}{s-\lambda_{n_x}} \end{bmatrix} M^{-1} B \quad (1.122)$$

Then by substitution for M and M^{-1} , one has

$$[G(s)] = C \left(\sum_{i=1}^{n_x} \frac{\vec{v}_i \vec{\mu}_i^T}{s - \lambda_i} \right) B \quad (1.123)$$

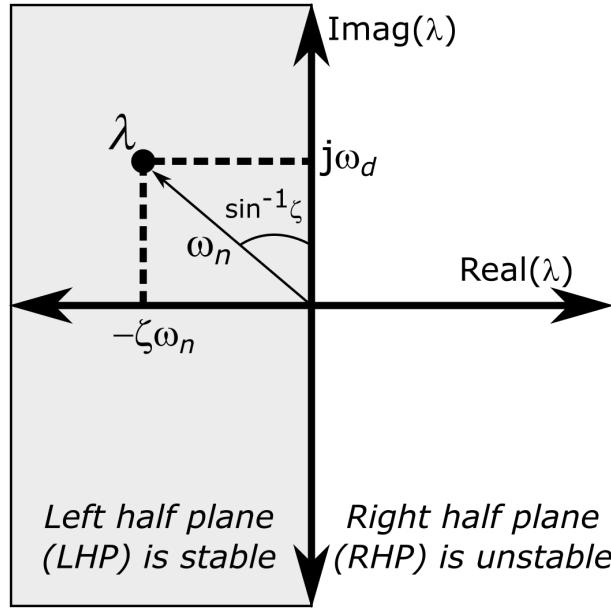
Furthermore, defining \vec{c}_j^T as the j^{th} row of C and \vec{b}_k as the k^{th} column of B , then for each input j and output k , one has the transfer function

$$G_{jk} = \sum_{i=1}^{n_x} \frac{\vec{c}_j^T \vec{v}_i \vec{\mu}_i^T \vec{b}_k}{(s - \lambda_i)} \quad (1.124)$$

which defines a partial-fraction decomposition.

The residues of these decompositions demonstrate that the i^{th} right eigenvector characterizes how the i^{th} mode affects the j^{th} output, while the i^{th} left eigenvalue characterize how the k^{th} input affects the *same* i^{th} modal state. Consequently, if any residue is zero, then either $\vec{\mu}_i^T \vec{b}_k = 0$ which indicates the k^{th} input does not affect the i^{th} mode, i.e. i is an **uncontrollable** mode, and/or $\vec{c}_j^T \vec{v}_i = 0$ which indicates the i^{th} mode does not affect the j^{th} output, i.e. i is an **unobservable** mode. Furthermore, if G_{jk} has an *approximate* pole-zero cancellation at $s = \lambda_i$, then the residue will be small and the contribution of the i^{th} mode will be relatively small on the output. It should be noted that if A is not diagonalizable, i.e. there are repeated eigenvalues, then, one must use Jordan matrices to form Λ , and a similar modal analysis can still be made with some caveats. These details are discussed in later parts of the textbook.

Lastly, it should be noted that often one desires to use a **reduced-order model (ROM)** for the purposes of simulation, control, and/or SID of dynamical systems. For LTI systems, one may use modal analysis to evaluate the most significant system effects in terms of the modes with significantly longer time constants, i.e. significantly smaller eigenvalues, as well as the higher gains, i.e. highest residues. A visual tool for quickly assessing LTI systems is the **pole-zero plot**, i.e. a plot of a transfer function's poles and zeros in the complex plane



which provides a visualization of the system stability, the slowest poles, and any approximate pole-zero cancellations.

References

For more information, please refer to the following

- Schmidt, D. K., “10.1 Linear System Analysis - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.2 State-Space Models,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 144-154
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.3 Transfer Function Models,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 155-170

1.4 Free, Impulse, and Step Responses of SISO LTI Systems

Consider a SISO LTI system represented by the standard LTI ODE

$$y^{[n]}(t) + a_{n-1}y^{[n-1]}(t) + \cdots + a_1\dot{y}(t) + a_0y(t) = b_mu^{[m]}(t) + \cdots + b_1\dot{u}(t) + b_0u(t) \quad (1.125)$$

with equivalent standard transfer function

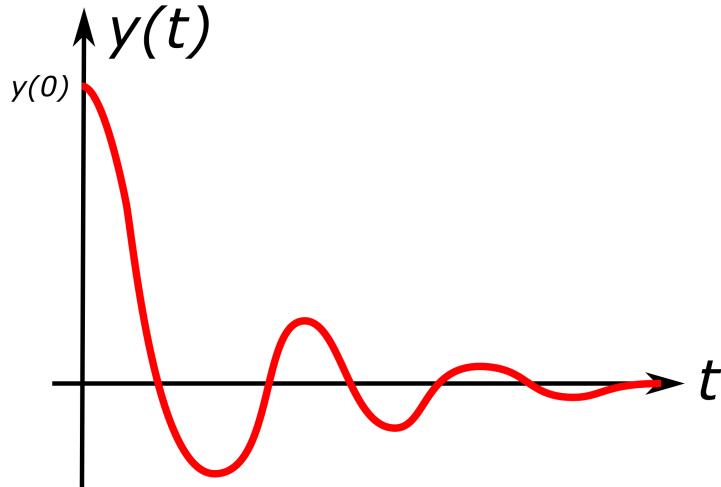
$$G(s) = \frac{b_ms^m + \cdots + b_1s + b_0}{s^n + a_{n-1}s^{n-1} + \cdots + a_1s + a_0} \quad (1.126)$$

This section will discuss two important system responses due to certain types of inputs for stable LTI systems.

The first is the **free response** given by $u(t)$ as

$$u(t) = 0 \quad \forall t \quad (1.127)$$

and $y(t)$ for boundary conditions given as initial conditions at $t = 0$, i.e. $(y(0), \dot{y}(0), y^{[n]}(0))$. Thus, this is also known as the **zero-input response** and the **initial condition response**. It also forms the **initial value problem (IVP)** for the ODE. In general, the initial condition response for stable SISO LTI systems has the form

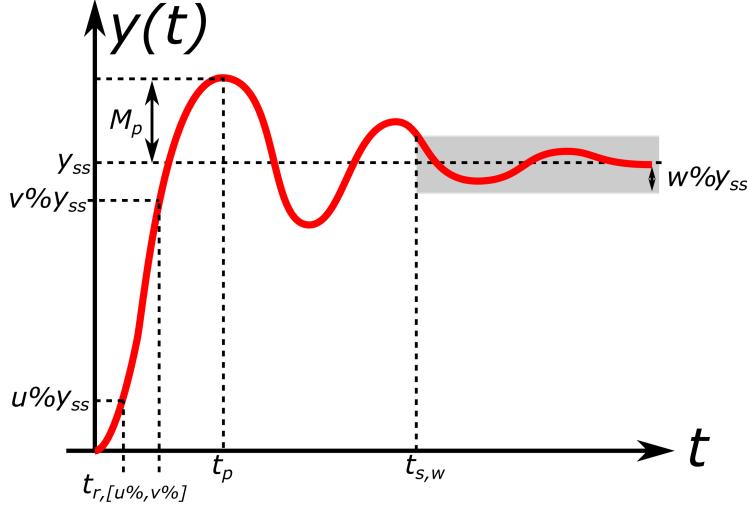


This plot primarily displays whether a system is stable or not, i.e. decays to zero or grows exponentially, while its other characteristics generally also depend on the specific initial conditions and the system poles.

The second and more useful response is the **step response** given by $y(0) = \dot{y}(0) = \dots = y^{[n]}(0) = 0$ and $u(t)$ is the **step input**, i.e.

$$u(t) = \begin{cases} 0 & \forall t < 0 \\ k & \forall t \geq 0 \end{cases} \quad (1.128)$$

where k is the **step amplitude**. If $k = 1$, this is called the **unit step**. In general, the step response for stable SISO LTI systems has the form



First, as $t \rightarrow \infty$, the stable LTI system will reach some **steady-state condition**, i.e. the output will become steady in value as all derivatives of the input and output are zero. This is summarized by the equation

$$y_{ss} = \frac{b_0}{a_0} u_{ss} \quad (1.129)$$

where $u_{ss} = k$ is the **steady-state input** signal, y_{ss} is the **steady-state output**, also known as the **final value**, and the ratio $\frac{b_0}{a_0}$ is the **steady-state gain**, which for a unit step is equivalent to the final value. Furthermore, as the initial conditions are given specifically as zero, the step response has other specific characteristics. The **rise time**, $t_{r,[u\%,v\%]}$, is defined as the time for the response to rise from $u\%$ of y_{ss} to $v\%$ of y_{ss} . Furthermore, the **peak time**, t_p , is the time for the response to reach its peak value and the corresponding percentage **maximum overshoot**, is given by

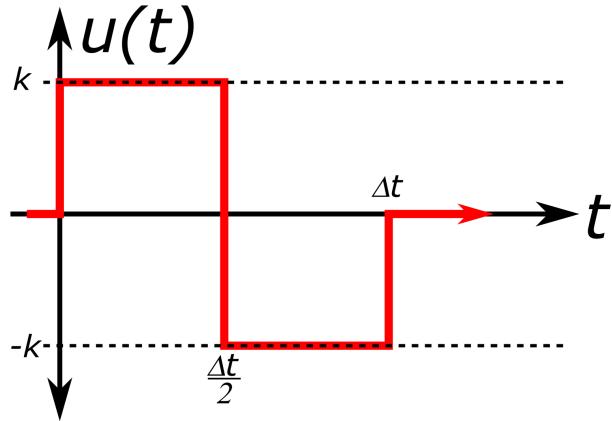
$$M_p = \frac{y(t_p) - y_{ss}}{y_{ss}} \quad (1.130)$$

Lastly, the **settling time**, $t_{s,w}$, is defined as the time for the response to reach and stay within $w\%$ of y_{ss} . The next section will derive equations for the unit step response characteristics for unimodal SISO LTI systems, i.e. first-order and underdamped second-order. Then, the qualitative effects of additional poles and zeros on the dominant mode unit step response will be discussed qualitatively.

As an aside, a related input signal to the step is the **doublet input**, i.e.

$$u(t) = \begin{cases} 0 & \forall t < 0 \\ k & \forall 0 \leq t < \frac{\Delta t}{2} \\ -k & \forall \frac{\Delta t}{2} \leq t < \Delta t \\ 0 & \forall t \geq \Delta t \end{cases} \quad (1.131)$$

where k is the **doublet amplitude** and Δt is the **doublet length**.



The primary benefit of using the doublet over the step response is that for stable systems, the output and the *integrated* output will return to zero.

Impulse Response

Similar to the initial condition response, consider the SISO LTI system response for an **impulse input**,

$$u(t) = \delta(t) \quad (1.132)$$

where $\delta(t)$ is the **Dirac delta** which can be “loosely” considered a function on the real line which is zero everywhere except at the origin, where it is infinite, but is also constrained to satisfy the identity, i.e.,

$$\delta(x) = \begin{cases} +\infty, & x = 0 \\ 0, & x \neq 0 \end{cases} \quad (1.133)$$

$$\int_{-\infty}^{\infty} \delta(x) dx = 1$$

However, this is merely a heuristic characterization as the Dirac delta is not a function in the traditional sense as no function defined on the real numbers has these properties. The Dirac delta function can be rigorously defined either as a distribution or as a measure, but an intuitive understanding is as the special continuous-time **impulse function**.

Then, by the Laplace transform one has

$$u(s) = \int_0^{\infty} \delta(t) e^{-st} dt \quad (1.134)$$

or by the definition of δ as zero everywhere except at $t = 0$ and its integral constraint as well as $e^{-s(0)} = 1$, one has

$$u(s) = 1 \quad (1.135)$$

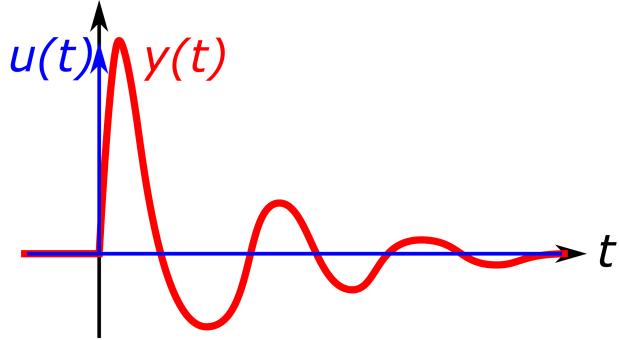
Thus,

$$y(s) = G(s) \quad (1.136)$$

or in the time domain by the inverse Laplace transform

$$y(t) = g(t) = \mathcal{L}^{-1}(G(s)) \quad (1.137)$$

where $g(t)$ is called the **impulse response**. Thus, by simulating an LTI in the time domain using an impulse input, one can completely determine the transfer function transformed to the time domain. An example of an impulse response plot is



which is somewhat similar to an initial condition response, except the excitation of the system occurs via the input and not the initial conditions which are assumed to be zero for the impulse.

Furthermore, consider the output response in the Laplace domain

$$y(s) = G(s)u(s) \quad (1.138)$$

which can be transformed to the time domain as the **convolution integral** between the impulse response and *any* input as

$$y(t) = \int_0^t g(\tau)u(t - \tau)d\tau \quad (1.139)$$

Thus, a type of SID method is known as **deconvolution** which uses measured output and input signals in the time domain to estimate the impulse response.

Unit Step Responses for Unimodal SISO LTI Systems

Consider a first-order LTI system with no zeros, i.e.

$$G(s) = \frac{b_0}{s + a_0} \quad (1.140)$$

Recalling the Laplace transform for a unit step, i.e. $u(s) = \frac{1}{s}$, the unit step response for a first-order LTI system in the Laplace domain is

$$y(s) = \frac{b_0}{s(s + a_0)} \quad (1.141)$$

which can be written in partial-fraction decomposition form as

$$y(s) = \frac{b_0}{a_0} \left(\frac{1}{s} + \frac{-1}{s + a_0} \right) \quad (1.142)$$

which, by the inverse Laplace transform, one has

$$y(t) = \frac{b_0}{a_0} (1 - e^{-a_0 t}) \quad (1.143)$$

which is an exponential decay from 0 to the steady-state output

$$y_{ss} = \frac{b_0}{a_0} \quad (1.144)$$

and does not have a peak or maximum overshoot.

To compute the rise and settling times, note that one can write

$$\frac{a_0 y(t)}{b_0} = 1 - e^{-a_0 t} \quad (1.145)$$

$$1 - \frac{a_0 y(t)}{b_0} = e^{-a_0 t} \quad (1.146)$$

and, by the natural logarithm, one has

$$t = -\frac{\ln \left(1 - \frac{a_0}{b_0} y(t)\right)}{a_0} \quad (1.147)$$

Thus, this system has a rise time from u to v given by

$$t_{r,[u,v]} = -\frac{\ln(1-u)}{a_0} + \frac{\ln(1-v)}{a_0} = \frac{v-u}{a_0} \quad (1.148)$$

and a $w\%$ settling time

$$t_{s,w} = -\frac{\ln w}{a_0} \quad (1.149)$$

where notably

$$t_{s,2\%} = \frac{3.9}{a_0} \quad (1.150)$$

and

$$t_{s,5\%} = \frac{3}{a_0} \quad (1.151)$$

Next, consider a stable second-order LTI system with no zeros, i.e.

$$G(s) = \frac{b_0}{s^2 + a_1 s + a_0} \quad (1.152)$$

which may be written in the alternate quadratic form as

$$G(s) = \frac{b_0}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (1.153)$$

where undamped natural frequency is

$$\omega_n = \sqrt{a_0} > 0 \quad (1.154)$$

and the damping ratio

$$\zeta = \frac{a_1}{2\omega_n} > 0 \quad (1.155)$$

has three different cases

1. **underdamped:** $0 < \zeta < 1$, i.e. one oscillatory decaying mode
2. **overdamped:** $\zeta > 1$, i.e. two exponentially decaying modes
3. **critically damped:** $\zeta = 1$, i.e. one exponentially decaying mode

thus, a unimodal system occurs for an underdamped second-order system.

Recalling the Laplace transform for a unit step, i.e. $u(s) = \frac{1}{s}$, the unit step response for an underdamped second-order LTI system in the Laplace domain is

$$y(s) = \frac{b_0}{s(s^2 + 2\zeta\omega_n s + \omega_n^2)} \quad (1.156)$$

which can be written in partial-fraction decomposition form as

$$y(s) = \frac{b_0}{\omega_n^2} \left(\frac{1}{s} + \frac{-s - \zeta\omega_n}{s^2 + 2\zeta\omega_n s + \omega_n^2} \right) \quad (1.157)$$

or, completing the square, one has

$$y(s) = \frac{b_0}{\omega_n^2} \left(\frac{1}{s} - \frac{s + \zeta\omega_n}{(s + \zeta\omega_n)^2 + \omega_n^2(1 - \zeta^2)} \right) \quad (1.158)$$

which, by the inverse Laplace transform, one has

$$y(t) = \frac{b_0}{\omega_n^2} \left[1 - e^{-\zeta\omega_n t} \cos(\omega_n \sqrt{1 - \zeta^2} t) \right] \quad (1.159)$$

or using the damped natural frequency, one has

$$y(t) = \frac{b_0}{\omega_n^2} \left[1 - e^{-\zeta\omega_n t} \cos(\omega_d t) \right] \quad (1.160)$$

is an oscillating exponential decay from 0 to the steady-state output

$$y_{ss} = \frac{b_0}{\omega_n^2} \quad (1.161)$$

It can be shown that the rise time from 0% to 100% is given by

$$t_{r,[0,1]} = \frac{\pi - \tan^{-1} \left(\frac{\sqrt{1-\zeta^2}}{\zeta} \right)}{\omega_d} \quad (1.162)$$

the peak time is given by

$$t_p = \frac{\pi}{\omega_d} \quad (1.163)$$

the maximum overshoot is given by

$$M_p = e^{-\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (1.164)$$

and the w% settling time

$$t_{s,w} = -\frac{\ln(w\sqrt{1 - \zeta^2})}{\zeta\omega_n} \quad (1.165)$$

Unit Step Responses for General SISO LTI Systems

For higher-order systems and system zeros, one can quantify the effects as follows. As an example of these effects, consider the nominal second-order LTI system, $G_1(s)$, as

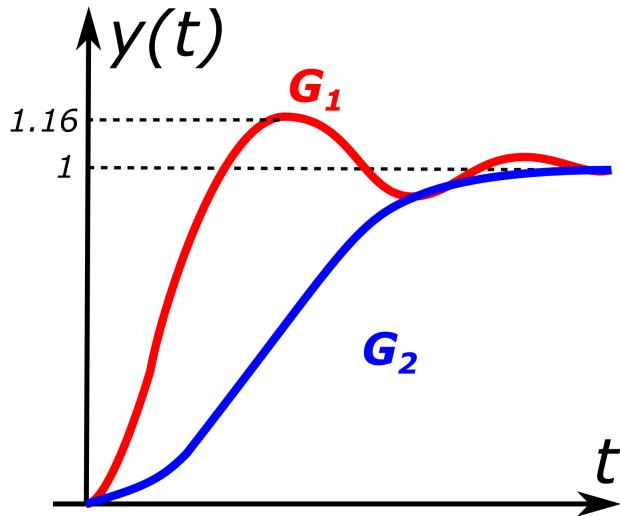
$$G_1(s) = \frac{4}{s^2 + 2s + 4} \quad (1.166)$$

Notably, G_1 is stable with poles at $-1 \pm j1.73$ and has steady-state output of $y_{ss} = \frac{b_0}{a_0} = \frac{4}{4} = 1$. It is underdamped with $\zeta = 0.5$ and $\omega_n = 2$ and has a maximum overshoot, $M_p \approx 0.16$.

Due to the principle of superposition, additional poles for a system simply add additional exponential terms to the general analytical solution, additional stable poles will generally increase the peak time and decrease the maximum overshoot from the nominal system. However, these effects are only significant if the additional poles are close in magnitude to the dominant poles, e.g. a factor of 4 is a good rule-of-thumb. As an example, consider an additional pole at $s = -1$ for the nominal system, then one has the third-order LTI system

$$G_2(s) = \frac{4}{s^2 + 2s + 4} \left(\frac{1}{s + 1} \right) \quad (1.167)$$

Plotting the unit step response for both $G_1(s)$ and $G_2(s)$, one has



One can explain the effects of a zero as follows. Let the nominal system be represented by

$$y_0(s) = G(s)u(s) \quad (1.168)$$

and let the new system, y_1 , have an additional zero, $s = z$, i.e.

$$y_1(s) = G(s)u(s) \left(\frac{-1}{z}s + 1 \right) \quad (1.169)$$

Then,

$$y_1(s) = \left(\frac{-1}{z}s + 1 \right) y_0(s) \quad (1.170)$$

and in the time domain

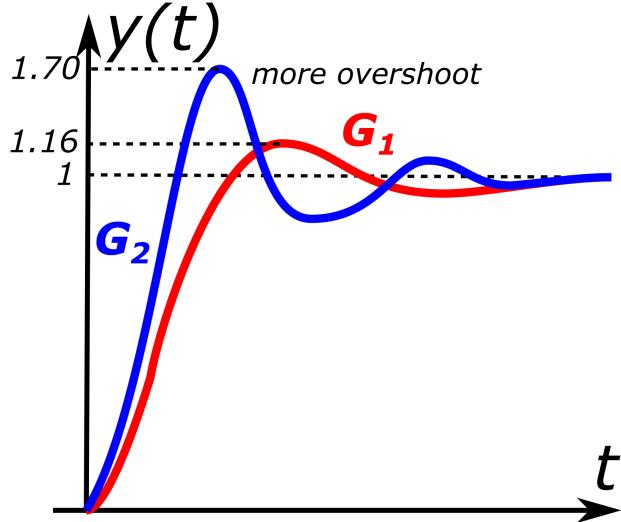
$$y_1(t) = \frac{-1}{z} \dot{y}_0(t) + y_0(t) \quad (1.171)$$

Thus, the step response consists of the nominal system response plus the “slope” of the nominal system response. Since the slope is initially positive for the nominal system response, one can infer that if $\text{Real}(z) < 0$, then there will be more maximum overshoot and if $\text{Real}(z) > 0$, then there will be also be an initial undershoot. However, these effects are only significant if the additional zeros are close in magnitude to the dominant poles, e.g. a factor of 4 is a good rule-of-thumb. If $\text{Real}(z) > 0$, then z is a **right half plane (RHP) zero**, otherwise if $\text{Real}(z) < 0$, z is a **left half plane (LHP) zero**.

As an example, consider a LHP zero at $s = -1$ for the nominal system, then one has the second-order LTI system

$$G_2(s) = \frac{4(s+1)}{s^2 + 2s + 4} \quad (1.172)$$

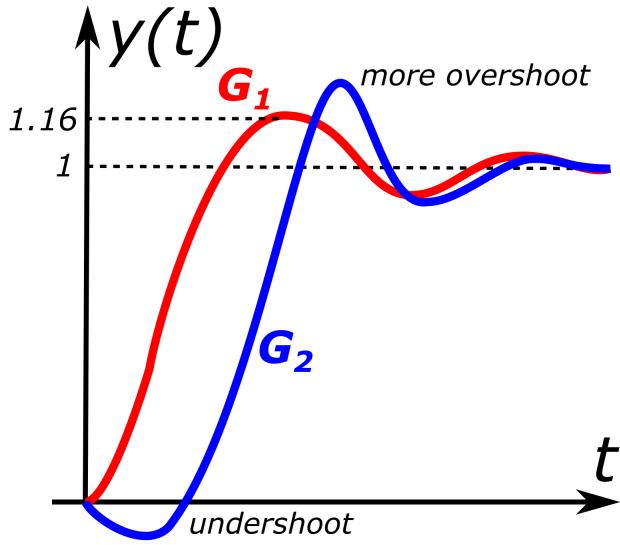
Plotting the unit step response for both $G_1(s)$ and $G_2(s)$, one has



Lastly, as a second example, consider a RHP zero at $s = 1$ for the nominal system, then one has the second-order LTI system

$$G_2(s) = \frac{4(-s+1)}{s^2 + 2s + 4} \quad (1.173)$$

Plotting the unit step response for both $G_1(s)$ and $G_2(s)$, one has



References

For more information, please refer to the following

- Nelson, R. C., “4.2 Second-Order Differential Equations,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 133-138
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.3 Transfer Function Models,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 168-170

1.5 Sinusoidal and Frequency Responses of SISO LTI Systems

Sinusoidal Response

Dynamical systems use signals as the time-dependent input and output variables. An important type of signal is a **periodic signal** which is one that repeats its sequence of values exactly after a fixed length of time, known as the **period**. As an example, consider the simplest periodic input signal, i.e. a sinusoid at frequency ω

$$u(t) = \cos(\omega t) \quad (1.174)$$

Then, using the convolution integral for the output, one has

$$y(t) = \int_0^t g(\tau) (\cos(\omega(t - \tau))) d\tau \quad (1.175)$$

or

$$y(t) = \int_0^\infty g(\tau) (\cos(\omega(t - \tau))) d\tau - \int_t^\infty g(\tau) (\cos(\omega(t - \tau))) d\tau \quad (1.176)$$

where, if the system is stable, the first term is the steady-state sinusoidal response and the second term is the **transient response** which decays with t .

Thus, any stable system output will reach the steady-state sinusoidal response given by

$$y_{sss}(t) = \int_0^{\infty} g(\tau) (\cos(\omega(t-\tau))) d\tau \quad (1.177)$$

which using Euler's formula for $\cos(\omega t)$ as

$$u(t) = \frac{1}{2} (e^{j\omega t} + e^{-j\omega t}) \quad (1.178)$$

one has

$$y_{sss}(t) = \frac{1}{2} \int_0^{\infty} g(\tau) e^{j\omega(t-\tau)} d\tau + \frac{1}{2} \int_0^{\infty} g(\tau) e^{-j\omega(t-\tau)} d\tau \quad (1.179)$$

or

$$y_{sss}(t) = \frac{1}{2} e^{j\omega t} \int_0^{\infty} g(\tau) e^{-j\omega\tau} d\tau + \frac{1}{2} e^{-j\omega t} \int_0^{\infty} g(\tau) e^{j\omega\tau} d\tau \quad (1.180)$$

and using the definition of the transfer function using the Laplace transform

$$y_{sss}(t) = \frac{1}{2} e^{j\omega t} G(j\omega) + \frac{1}{2} e^{-j\omega t} G(-j\omega) \quad (1.181)$$

or finally, the **steady-state sinusoidal response**

$$y_{sss}(t) = \text{Real}\{G(j\omega)\} \cos(\omega t) - \text{Imag}\{G(j\omega)\} \sin(\omega t) \quad (1.182)$$

which can alternatively be written in **polar form** as

$$y_{sss}(t) = |G(j\omega)| \cos(\omega t + \angle G(j\omega)) \quad (1.183)$$

where the **magnitude**, also known as the **gain**, is defined as

$$|G(j\omega)| = \sqrt{\text{Real}\{G(j\omega)\}^2 + \text{Imag}\{G(j\omega)\}^2} \quad (1.184)$$

and the **phase** is defined as

$$\angle G(j\omega) = \tan^{-1} \left(\frac{\text{Imag}\{G(j\omega)\}}{\text{Real}\{G(j\omega)\}} \right) \quad (1.185)$$

This derivation demonstrates that the steady-state sinusoidal response is a sinusoid at the same frequency, but with a different magnitude, and phase dependent on the transfer function evaluated at $s = j\omega$, i.e. $G(j\omega)$.

Frequency Response

The importance of this substitution can be formally stated by the definition of the **Fourier transform** defined as

$$f(\omega) = \mathcal{F}\{f(t)\} = \int_{-\infty}^{\infty} f(t) e^{-j\omega t} dt \quad (1.186)$$

where ω is the **angular frequency**, typically given in radians/second. Sometimes the **periodic frequency**, ξ , is used where $\omega = 2\pi\xi$ and typically given in Hertz (Hz) or cycles/second. One intuitive way to understand the Fourier transform is through the **Fourier series** which states that a *periodic* function can be rewritten as the infinite summation of harmonically related sinusoids, similar to the Taylor series of arbitrary functions as infinite summations of polynomial terms. Then, as the “period” of any periodic function is allowed to approach infinity, i.e. it does not repeat in finite time, the Fourier series becomes the Fourier transform.

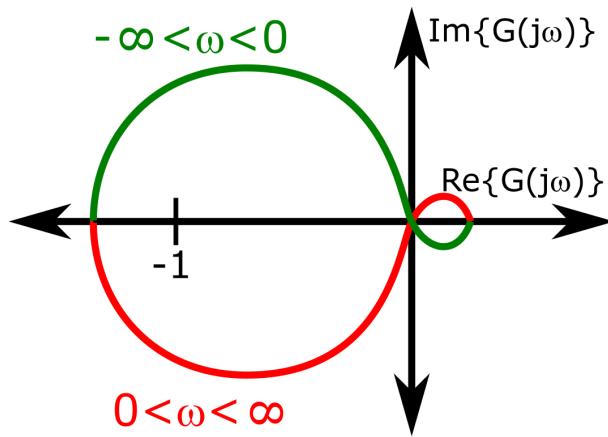
This transformation is useful through the **Fourier inversion theorem** which states that for “well-behaved” functions, e.g. LTI ODEs, it is possible to recover the function entirely from its Fourier transform using the **inverse Fourier transform** defined as

$$f(t) = \mathcal{F}^{-1}\{f(\omega)\} = \frac{1}{2\pi} \int_{-\infty}^{\infty} f(\omega)e^{j\omega t} d\omega \quad (1.187)$$

Notably, the Fourier and inverse Fourier transforms are exactly equivalent to the Laplace transforms with the substitution $s = j\omega$.

Thus, the Fourier transform of a SISO LTI system, $G(j\omega)$, considered as a continuous function of all $\omega \in [0, \infty)$ contains all responses of the SISO LTI system as a function of frequency, i.e. it models the **frequency response** of the SISO LTI system. With the Fourier inversion theorem in mind, the frequency response provides the SISO LTI system response to *all* harmonic sinusoids within any “well-behaved” input signal and from which one can exactly replicate the output signal. Notably, the class of well-behaved signals also includes the step signal, in fact, the *steady-state* output of the step response is equivalent to the frequency response at $\omega = 0$. Thus, the frequency response also provides the magnitude and phase values for all steady-state step and sinusoidal responses. Lastly, the frequency response can also be regarded as the Fourier transform of the impulse response, i.e. $g(t) \rightarrow G(j\omega)$ for all $\omega \in [0, \infty)$. Two common plots used to analyze the frequency response of SISO LTI systems are the Nyquist and Bode plots. For MIMO LTI systems, one must employ multivariate frequency domain analysis which is discussed later in this textbook.

Similar to the Bode plot, the **Nyquist plot** can be used to analyze the frequency response of a transfer function, i.e. $G(s)$ with $s = j\omega$. However, opposed to the Bode plot, the Nyquist plot visualizes the real and imaginary parts of $G(j\omega)$ as a single curve with the real part on the horizontal axis and the imaginary on the vertical axis. It should also be noted the convention for the Nyquist plot is to plot over $-\infty < \omega < \infty$ as opposed to $\omega \geq 0$ for the Bode plot. This convention results in a reflected curve about the real axis for $\omega < 0$ with respect to $\omega > 0$ as a transfer function value $G(j\omega) = \alpha + j\beta$ simply provides the complex conjugate for $\omega < 0$ with respect to $\omega > 0$. An example of a Nyquist plot is the following:



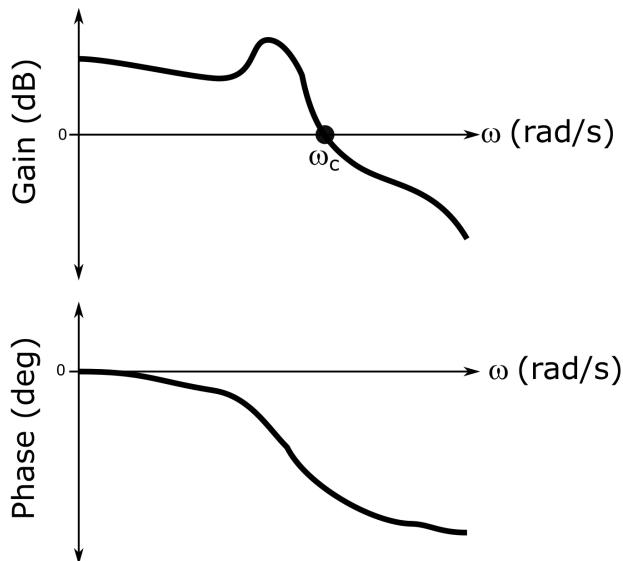
However, this textbook will focus on the Bode plot for analysis and design.

Bode Plot

The **Bode plot** consists of two subplots:

1. the magnitude, also known as the gain, $|G(j\omega)|$ (vertical) versus ω (horizontal)
2. the phase $\angle G(j\omega)$ (vertical) versus ω (horizontal)

For the Bode plot, it is common to plot the frequency, ω , on a \log_{10} scale in radians/sec, the phase, $\angle G(j\omega)$, in degrees ($^\circ$), and the magnitude, $|G(j\omega)|$, in units of decibels (dB), i.e. $20 \log_{10} |G(j\omega)|$. In this manner, **crossover frequencies**, ω_c , occur for $|G(j\omega_c)|_{dB} = 0$ dB An example of a Bode plot is the following:



Some useful conversions for decibels to gain are as follows.

| Decibels (dB) | Gain |
|---------------|----------------------|
| 40 dB | 100 |
| 20 dB | 10 |
| 6 dB | 2 |
| 3 dB | $\sqrt{2}$ |
| 0 dB | 1 |
| -3 dB | $\frac{1}{\sqrt{2}}$ |
| -6 dB | $\frac{1}{2}$ |
| -20 dB | $\frac{1}{10}$ |
| -40 dB | $\frac{1}{100}$ |

An important characteristic of the Bode plot, is that for multiplying complex numbers in polar form, e.g. $n_1 = A_1 e^{j\phi_1}$ and $n_2 = A_2 e^{j\phi_2}$, one has

$$n_1 n_2 = A_1 A_2 e^{j(\phi_1 + \phi_2)} \quad (1.188)$$

which is a complex number with multiplicative magnitude, i.e. $|A_1 A_2|$, and additive phase, i.e. $\phi_1 + \phi_2$. However, if the magnitude is in dB, then one has

$$20 \log_{10} |A_1 A_2| = 20 \log_{10} |A_1| + 20 \log_{10} |A_2| \quad (1.189)$$

which is additive in dB, i.e.

$$|A_1 A_2|_{\text{dB}} = |A_1|_{\text{dB}} + |A_2|_{\text{dB}} \quad (1.190)$$

Recalling that SISO LTI systems can be rewritten into its poles and zeros, i.e.

$$G(s) = K \frac{(s - z_1) \cdots (s - z_m)}{(s - p_1) \cdots (s - p_n)} \quad (1.191)$$

the Bode plot for any SISO LTI system can be constructed by *adding* the Bode plots for each real and complex-conjugate pole and zero assuming the magnitude is in dB. As such, this section will discuss the Bode plots for differentiators, integrators, first-order zeros, first-order poles, underdamped second-order zeros, and underdamped second-order poles. This section will also discuss the **straight-line approximations** which provide a quick method to sketch Bode plots. From these plots and the previous additive property, one can construct Bode plots and sketch straight-line approximations for any SISO LTI system.

As an aside, this additive property also infers that if given the Bode plot of the frequency response, one could estimate the poles and zeros of an unknown SISO LTI system, i.e. the entire transfer function. This idea qualitatively forms the basis for frequency domain-based SID.

The **DC gain** of the Bode plot is the magnitude at $\omega = 0$, i.e.

$$|G(0)| = \frac{k z_1 \cdots z_m}{p_1 \cdots p_n} \quad (1.192)$$

This terminology comes from the origin of the Bode plot for frequency domain analysis of electrical systems using direct current (DC) and alternating current (AC) components. Note that the DC gain is also the steady-state gain for the step response of stable LTI systems.

The **asymptotic slope** of the Bode plot can be calculated by analyzing the limit of the gain asymptotically, i.e.

$$\lim_{\omega \rightarrow \infty} |G(j\omega)|_{\text{dB}} = 20 \log_{10} \omega^{m-n} \quad (1.193)$$

Next, consider increasing ω by a factor of 10, i.e. a **decade** in frequency, and note

$$20 \log_{10}(10\omega)^x = 20 \log_{10} \omega^x + 20 \log_{10} 10^x = 20 \log_{10} \omega^x + 20x \quad (1.194)$$

Thus, as one moves up a decade in frequency, the asymptotic slope is

$$\left(\frac{d|G(j\omega)|_{\text{dB}}}{d(10\omega)} \right)_{\omega \rightarrow \infty} = 20(m-n) \quad (1.195)$$

Lastly, it should be noted that one must also account for the gain $k \neq 1$. If $k < 0$, then the phase will have $\pm 180^\circ$ added. If $|k| > 1$, the magnitude will move up k_{dB} , while if $|k| < 1$, the magnitude will move down $-|k|_{\text{dB}}$.

References

For more information, please refer to the following

- Nelson, R. C., “4.2 Second-Order Differential Equations,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 133-138
- Nelson, R. C., “Appendix D: Review of Control System Analysis Techniques,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 439-434
- Schmidt, D. K., “10.1 Linear System Analysis - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.3 Transfer Function Models,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 155-170

1.6 Frequency Responses of Fundamental SISO LTI Systems

Frequency Response of Pure Differentiator

Consider a SISO LTI system with a transfer function of a **pure differentiator**

$$G(s) = s \quad (1.196)$$

which has a zero at $s = 0$ and a DC gain of $-\infty$ dB. The frequency response $G(j\omega)$ can be computed as

$$G(j\omega) = j\omega \quad (1.197)$$

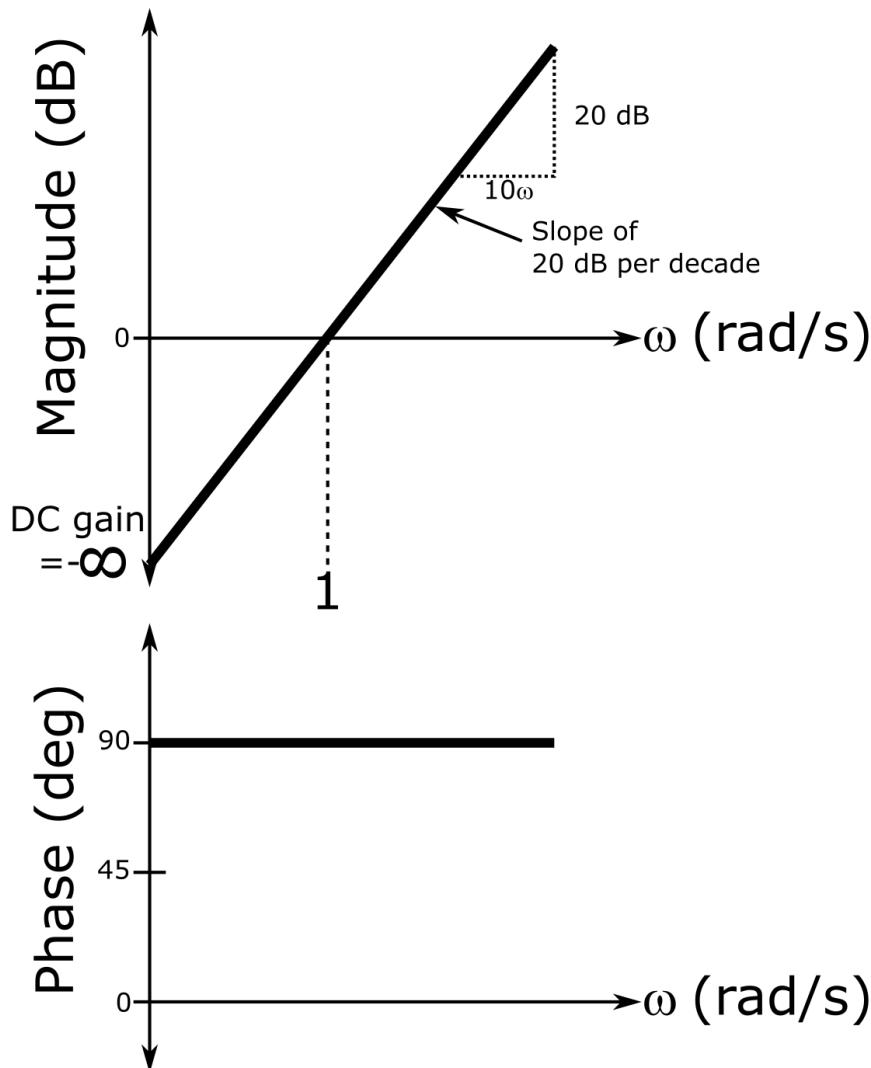
The magnitude can be computed as

$$|G(j\omega)|_{\text{dB}} = 20 \log_{10} |G(j\omega)| = 20 \log_{10} \omega \quad (1.198)$$

The phase can be computed as

$$\angle G(j\omega) = 90^\circ, \quad \forall \omega \quad (1.199)$$

Plotting, one has



Frequency Response of Pure Integrator

Consider a SISO LTI system with a transfer function of a **pure integrator**

$$G(s) = \frac{1}{s} \quad (1.200)$$

which has a pole at $s = 0$ and a DC gain of $+\infty$ dB. The frequency response $G(j\omega)$ can be computed as

$$G(j\omega) = \frac{1}{j\omega} = -j\frac{1}{\omega} \quad (1.201)$$

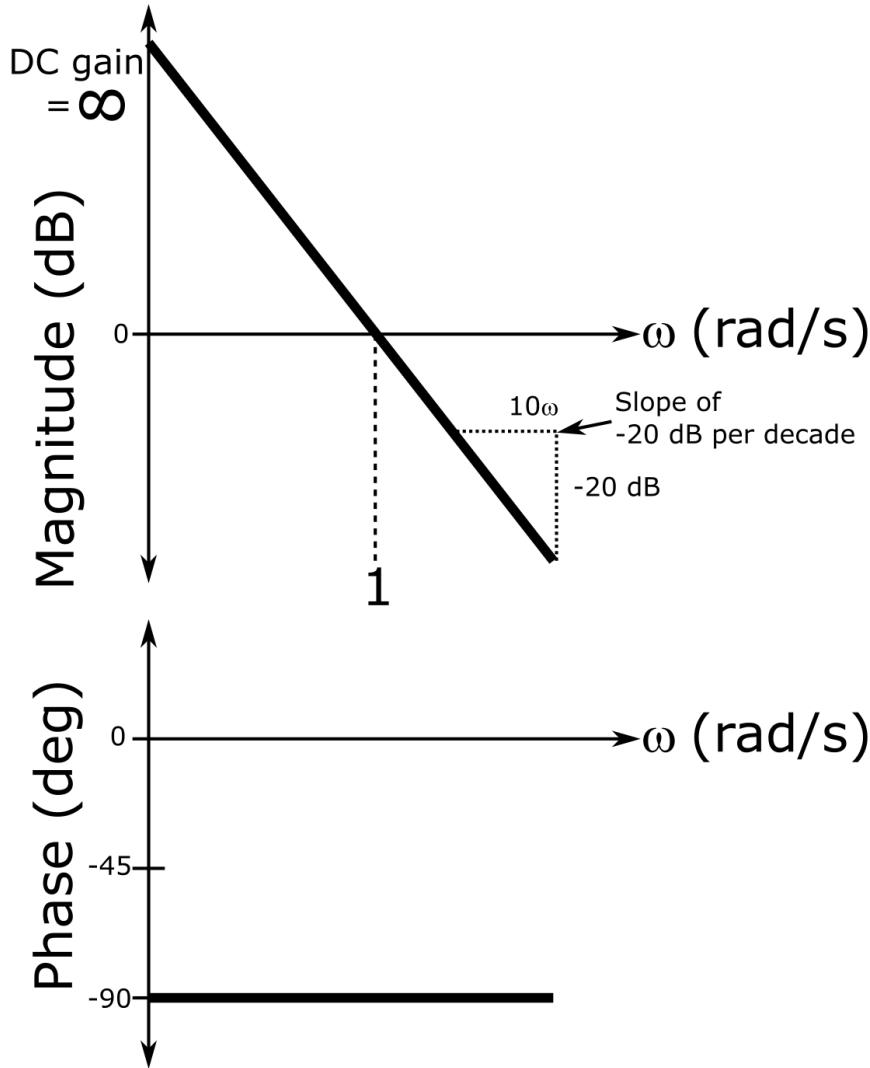
The magnitude can be computed as

$$|G(j\omega)|_{\text{dB}} = 20 \log_{10} |G(j\omega)| = 20 \log_{10} \frac{1}{\omega} = -20 \log_{10} \omega \quad (1.202)$$

The phase can be computed as

$$\angle G(j\omega) = -90^\circ, \quad \forall \omega \quad (1.203)$$

Plotting, one has



Frequency Response of First-Order Zero

Consider a SISO LTI system with a transfer function of a first-order zero, i.e.

$$G(s) = s - z \quad (1.204)$$

which has a zero at $s = z$, a DC gain of $20 \log_{10} |z|$, and an asymptotic slope of 20 dB per decade. The frequency response can be computed as

$$G(j\omega) = -z + j\omega \quad (1.205)$$

The magnitude can be computed as

$$|G(j\omega)|_{\text{dB}} = 20 \log_{10} |G(j\omega)| = 20 \log_{10} \left(\sqrt{z^2 + \omega^2} \right) \quad (1.206)$$

where notably, at $\omega = |z|$, one has

$$|G(j|z|)|_{\text{dB}} = 20 \log_{10} |z| \sqrt{2} = 20 \log_{10} |z| + 3 \text{ dB} \quad (1.207)$$

or the DC gain plus 3 dB.

The phase can be computed as

$$\angle G(j\omega) = \tan^{-1} \left[\frac{\omega}{-z} \right] \quad (1.208)$$

where notably

$$\angle G(j\omega) \rightarrow 90^\circ \text{ as } \omega \rightarrow \infty \quad (1.209)$$

for $z > 0$

$$\angle G(j0) = \pm 180^\circ \quad (1.210)$$

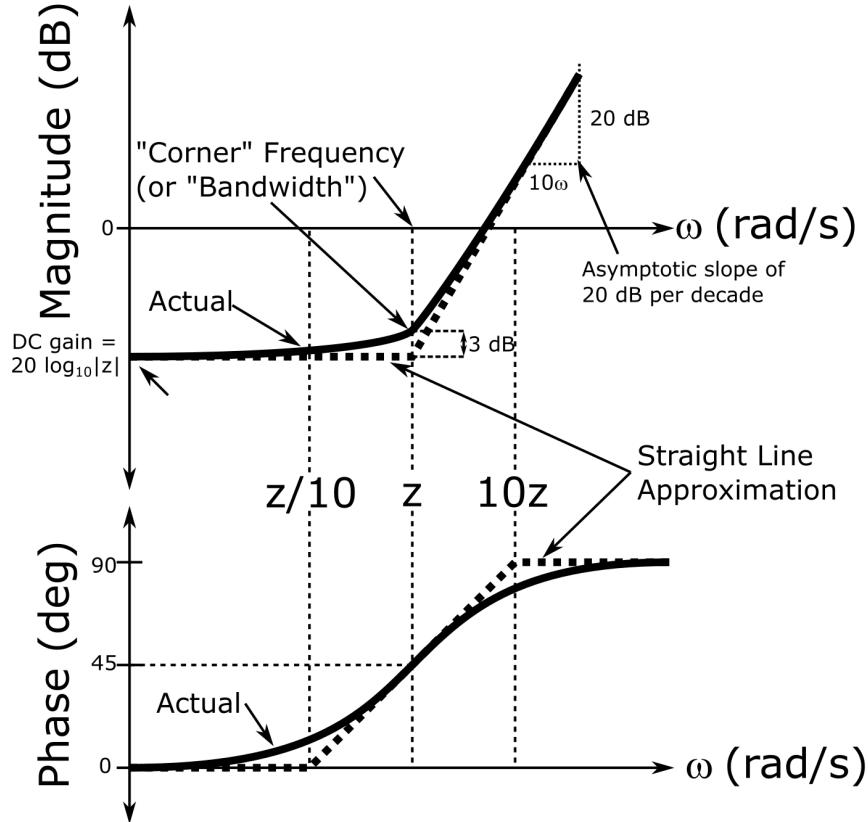
$$\angle G(jz) = 135^\circ \quad (1.211)$$

and for $z < 0$

$$\angle G(j0) = 0^\circ \quad (1.212)$$

$$\angle G(-jz) = 45^\circ \quad (1.213)$$

Plotting for $-1 < z < 0$, one has



Frequency Response of First-Order Pole

Consider a SISO LTI system with a transfer function of a first-order pole, i.e.

$$G(s) = \frac{1}{s - p} \quad (1.214)$$

which has a pole at $s = p$, a DC gain of $-20 \log_{10} |p|$, and an asymptotic slope of -20 dB per decade.

The frequency response $G(j\omega)$ can be computed as

$$G(j\omega) = \frac{1}{j\omega - p} \quad (1.215)$$

$$G(j\omega) = \frac{1}{j\omega - p} \frac{-p - j\omega}{-p - j\omega} \quad (1.216)$$

$$G(j\omega) = \frac{-p}{p^2 + \omega^2} - j \frac{\omega}{p^2 + \omega^2} \quad (1.217)$$

The magnitude can be computed as

$$|G(j\omega)|_{\text{dB}} = 20 \log_{10} |G(j\omega)| = 20 \log_{10} \frac{1}{\sqrt{p^2 + \omega^2}} \quad (1.218)$$

where notably

$$|G(j|z|)|_{\text{dB}} = 20 \log_{10} |z| \sqrt{2} = -20 \log_{10} |p| - 3 \text{ dB} \quad (1.219)$$

or the DC gain minus 3 dB.

The phase can be computed as

$$\angle G(j\omega) = \tan^{-1} \left[\frac{\frac{-\omega}{p^2 + \omega^2}}{\frac{p}{p^2 + \omega^2}} \right] \quad (1.220)$$

$$\angle G(j\omega) = \tan^{-1} \left[\frac{-\omega}{p} \right] \quad (1.221)$$

where notably

$$\angle G(j\omega) \rightarrow -90^\circ \text{ as } \omega \rightarrow \infty \quad (1.222)$$

for $p > 0$

$$\angle G(j0) = \pm 180^\circ \quad (1.223)$$

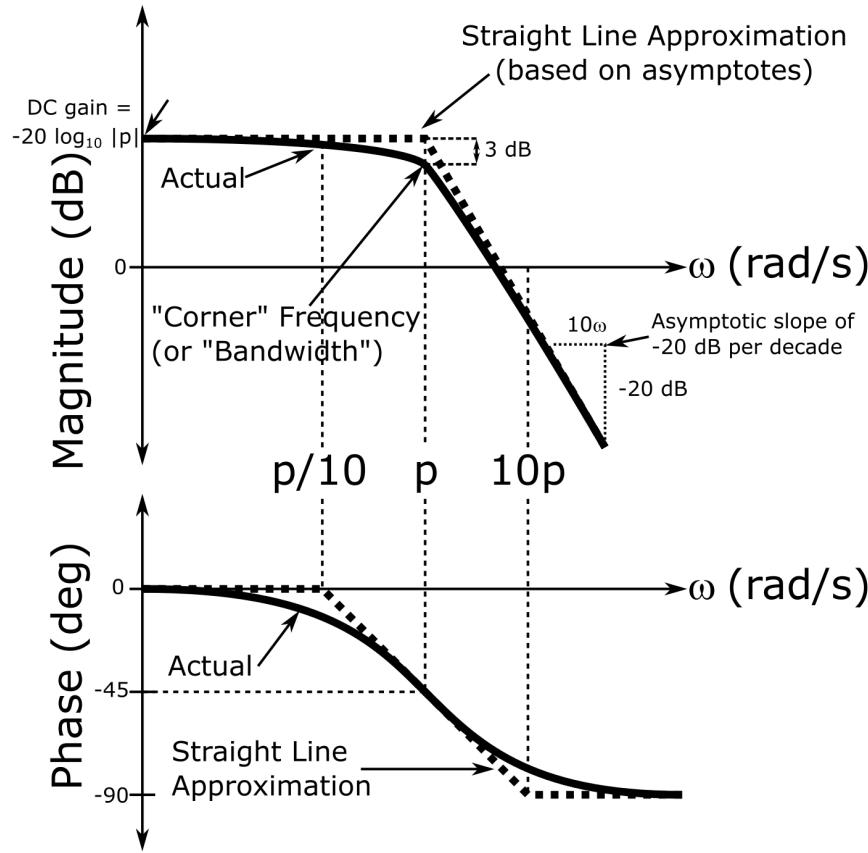
$$\angle G(jp) = -135^\circ \quad (1.224)$$

and for $p < 0$

$$\angle G(j0) = 0^\circ \quad (1.225)$$

$$\angle G(-jp) = -45^\circ \quad (1.226)$$

Plotting for $-1 < p < 0$, i.e. a stable transfer function, one has



Frequency Response for Underdamped Second-Order Zero

Consider a SISO LTI system with a transfer function of an underdamped second-order zero, i.e.

$$G(s) = s^2 + 2\zeta\omega_n s + \omega_n^2 \quad (1.227)$$

which has zeros at $s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$, a DC gain of $20 \log_{10} \omega_n^2$, and an asymptotic slope of 40 dB per decade.

The frequency response can be computed as

$$G(j\omega) = -\omega^2 + 2j\zeta\omega_n\omega + \omega_n^2 \quad (1.228)$$

$$G(j\omega) = (\omega_n^2 - \omega^2) + 2j\zeta\omega_n\omega \quad (1.229)$$

The magnitude can be computed as

$$|G(j\omega)|_{\text{dB}} = 20 \log_{10} |G(j\omega)| = 20 \log_{10} \left(\sqrt{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2} \right) \quad (1.230)$$

where notably

$$|G(j|\omega_n|)|_{\text{dB}} = 20 \log_{10} \left(\sqrt{4\zeta^2\omega_n^4} \right) \quad (1.231)$$

$$|G(j\omega_n)|_{\text{dB}} = 20 \log_{10} \omega_n^2 + 20 \log_{10} 2\zeta \quad (1.232)$$

or the DC gain plus some amount dependent on ζ . For underdamped zeros, i.e. $0 < \zeta < 1$, one has

$$|G(j\omega_n)|_{\text{dB}} < 20 \log_{10} \omega_n^2 + 20 \log_{10} 2(1) \approx 20 \log_{10} \omega_n^2 + 6 \text{dB} \quad (1.233)$$

and for $\zeta < 0.5$, $|G(j\omega_n)|_{\text{dB}} < 20 \log_{10} \omega_n^2$. Furthermore, if $\zeta < \frac{1}{\sqrt{2}} = 0.707$, then it can be proven using calculus that at $\omega = \omega_n \sqrt{1 - 2\zeta^2}$ there will be a **resonant valley**. This valley becomes more pronounced for smaller damping ratios and is significant for $\zeta < 0.5$.

The phase can be computed as

$$\angle G(j\omega) = \tan^{-1} \left[\frac{2\zeta\omega_n\omega}{\omega_n^2 - \omega^2} \right] \quad (1.234)$$

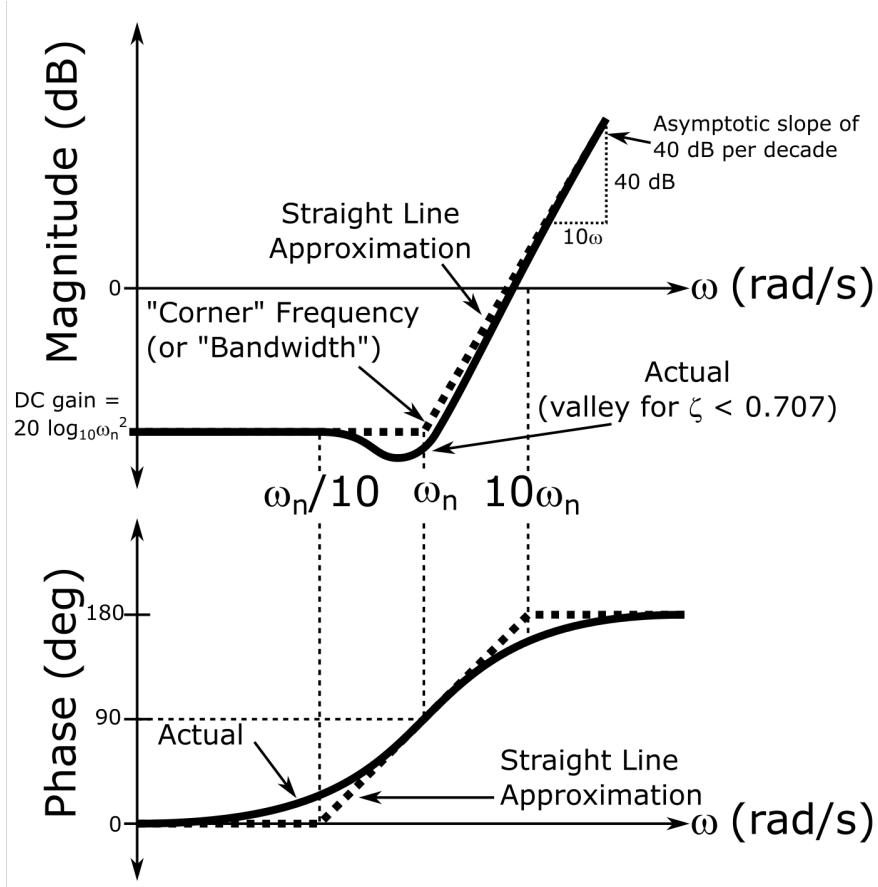
where notably

$$\angle G(j\omega_n) = 90^\circ \quad (1.235)$$

and

$$\angle G(j\omega) \rightarrow \pm 180^\circ \text{ as } \omega \rightarrow \infty \quad (1.236)$$

Plotting for $0 < \omega_n^2 < 1$ and $0 < \zeta < 1$, one has



Frequency Response for Underdamped Second-Order Pole

Consider a SISO LTI system with a transfer function of an underdamped second-order pole, i.e.

$$G(s) = \frac{1}{s^2 + 2\zeta\omega_n s + \omega_n^2} \quad (1.237)$$

which has poles at $s = -\zeta\omega_n \pm \omega_n\sqrt{\zeta^2 - 1}$, a DC gain of $-20 \log_{10} \omega_n^2$, and an asymptotic slope of -40 dB per decade.

The frequency response can be computed as

$$G(j\omega) = \frac{1}{-\omega^2 + 2j\zeta\omega_n\omega + \omega_n^2} \quad (1.238)$$

$$G(j\omega) = \frac{1}{\omega_n^2 - \omega^2 + 2j\zeta\omega_n\omega} \frac{\omega_n^2 - \omega^2 - 2j\zeta\omega_n\omega}{\omega_n^2 - \omega^2 - 2j\zeta\omega_n\omega} \quad (1.239)$$

$$G(j\omega) = \frac{\omega_n^2 - \omega^2}{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2} - j \frac{2\zeta\omega_n\omega}{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2} \quad (1.240)$$

The magnitude can be computed as

$$|G(j\omega)|_{\text{dB}} = 20 \log_{10} |G(j\omega)| = 20 \log_{10} \frac{1}{\sqrt{(\omega_n^2 - \omega^2)^2 + 4\zeta^2\omega_n^2\omega^2}} \quad (1.241)$$

where notably

$$|G(j|\omega_n|)|_{\text{dB}} = 20 \log_{10} \frac{1}{\sqrt{4\zeta^2\omega_n^4}} \quad (1.242)$$

$$|G(j|\omega_n|)|_{\text{dB}} = -20 \log_{10} \omega_n^2 - 20 \log_{10} 2\zeta \quad (1.243)$$

or the DC gain minus some amount dependent on $\zeta > 0$. For underdamped zeros, i.e. $0 < \zeta < 1$, one has

$$|G(j|\omega_n|)|_{\text{dB}} > 20 \log_{10} \omega_n^2 - 20 \log_{10} 4 \approx 20 \log_{10} \omega_n^2 - 6 \text{dB} \quad (1.244)$$

and for $\zeta < 0.5$, $|G(j|\omega_n|)|_{\text{dB}} > 20 \log_{10} \omega_n^2$. Furthermore, if $\zeta < \frac{1}{\sqrt{2}} = 0.707$, then it can be proven using calculus that at $\omega = \omega_n\sqrt{1 - 2\zeta^2}$ there will be a **resonant peak** of magnitude where this peak becomes more pronounced for small damping ratios and is significant for $\zeta < 0.5$.

The phase can be computed as

$$\angle G(j\omega) = \tan^{-1} \left[\frac{-2\zeta\omega_n\omega}{\omega_n^2 - \omega^2} \right] \quad (1.245)$$

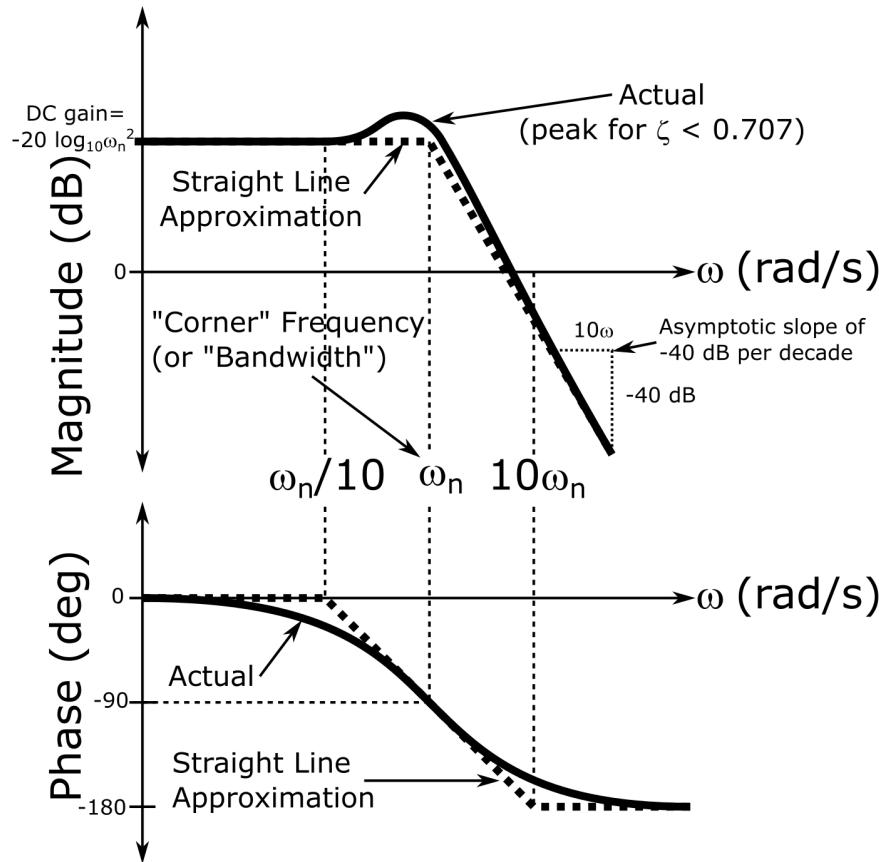
where notably

$$\angle G(j\omega_n) = -90^\circ \quad (1.246)$$

and

$$\angle G(j\omega) \rightarrow \pm 180^\circ \text{ as } \omega \rightarrow \infty \quad (1.247)$$

Plotting for $0 < \omega_n^2 < 1$ and $0 < \zeta < 1$, one has



References

For more information, please refer to the following

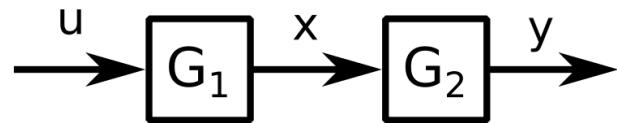
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.3 Transfer Function Models,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 155-170

Continuous-Time SISO LTI Feedback Control Theory

2.1 Introduction to Control Systems

Interconnections of SISO LTI Systems

For model-based design of SISO LTI control systems, one must understand some of the block diagram algebra for interconnected SISO LTI systems. For building an overall system transfer function, it is useful to consider the input and output to various interconnected systems represented by transfer functions, in particular the serial/cascade, parallel, and feedback interconnections. A **serial interconnection**, also known as a **cascade interconnection** can be represented as



and in the Laplace domain

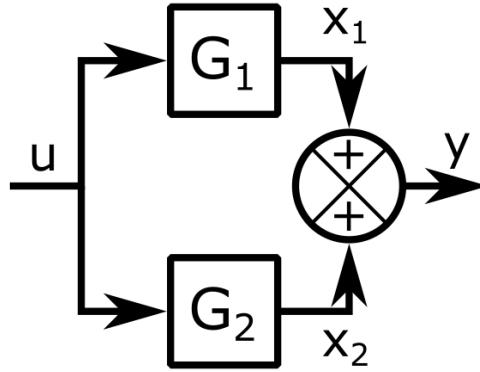
$$y(s) = G_2(s)x(s) \quad (2.1)$$

$$x(s) = G_1(s)u(s) \quad (2.2)$$

or for an overall system transfer function

$$y(s) = G(s)u(s) = [G_2(s)G_1(s)]u(s) \quad (2.3)$$

A **parallel interconnection** can be represented as



and in the Laplace domain

$$x_1(s) = G_1(s)u(s) \quad (2.4)$$

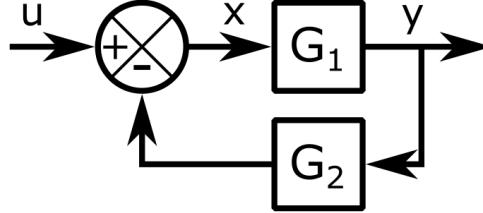
$$x_2(s) = G_2(s)u(s) \quad (2.5)$$

$$y(s) = x_1(s) + x_2(s) \quad (2.6)$$

or for an overall system transfer function

$$y(s) = G(s)u(s) = [G_1(s) + G_2(s)]u(s) \quad (2.7)$$

A (negative) **feedback interconnection** can be represented as



and in the Laplace domain

$$y(s) = G_1(s)x(s) \quad (2.8)$$

$$x(s) = u(s) - G_2(s)y(s) \quad (2.9)$$

$$y(s) = G_1(s)(u(s) - G_2(s)y(s)) \quad (2.10)$$

$$(1 + G_1(s)G_2(s))y(s) = G_1(s)u(s) \quad (2.11)$$

or for an overall system transfer function

$$y(s) = G(s)u(s) = \left[\frac{G_1(s)}{1 + G_2(s)G_1(s)} \right] u(s) \quad (2.12)$$

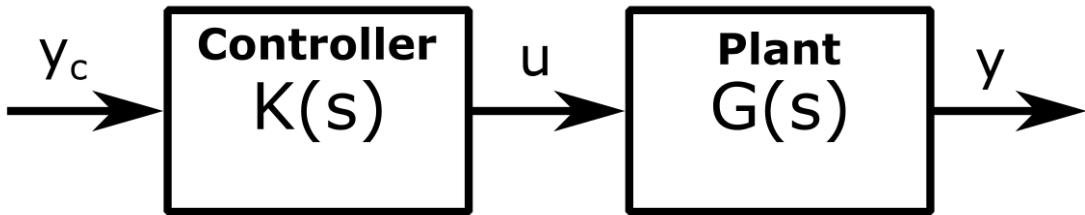
Lastly, when forming transfer functions for interconnected systems, it is important to take note of any pole-zero cancellations as this may result in an unstable feedback system due to the dynamics of interior signals of the system.

Open- and Closed-Loop Control Laws

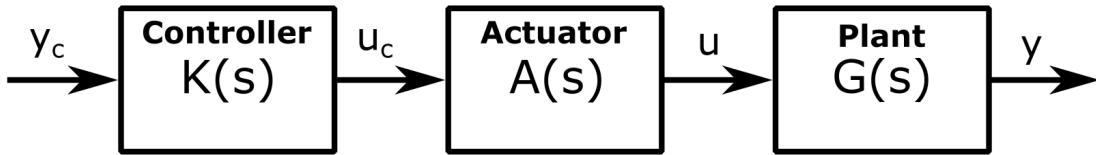
The simplest types of control laws are **open-loop control** strategies, also known as **feedforward control**. In general, this logically flows as the following

1. the user specifies a commanded output y_c
2. the control law sets the input u as a function of that commanded output
3. this dynamically changes the output $y(t)$

For the open-loop control of SISO LTI systems, one would have a simple system block diagram as



For real systems, u will undergo some dynamics itself, i.e. the plant input won't instantaneously reach the electric controller's computed u , but is actuated using some electro-mechanical interface which one can represent by an actuation system or **actuator**. This additional consideration would result in the following block diagram for SISO LTI systems.



Using dynamical systems theory, one can typically model actuation systems for vehicles, $A(s)$, as first- or second-order LTI systems, e.g.

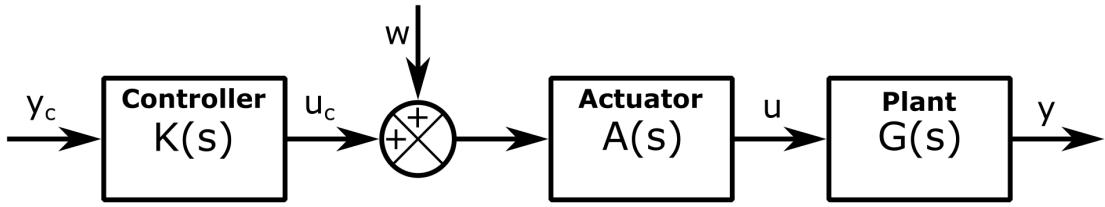
$$A(s) = \frac{\omega_a}{s + \omega_a} \quad (2.13)$$

or

$$A(s) = \frac{\omega_a^2}{s^2 + 2\zeta_a\omega_a s + \omega_a^2} \quad (2.14)$$

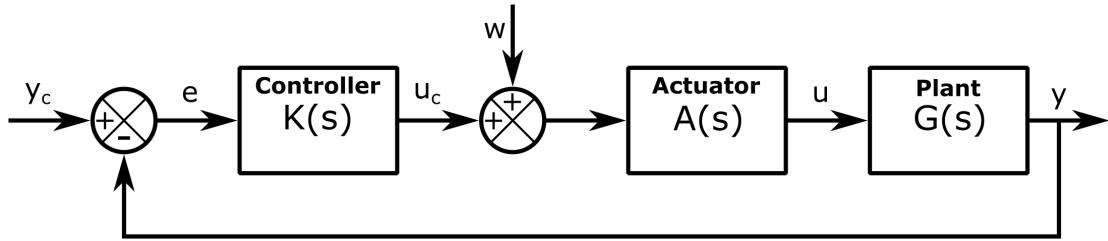
where ω_a is the bandwidth of the actuator in rad/s and ζ_a is the damping. Actuators also typically have hard limits on the minimum and maximum output as well as hard rate limits.

However, one consideration for real systems is the fact that one will have additional disturbances to the plant model. These effects can be modeled as an additive disturbance signal, w , to the control input, u , which results in a block diagram for SISO LTI systems as



Thus, if one does not have explicit knowledge of these disturbances, then one may not be able to guarantee the control system satisfies the design requirements using only open-loop control as this strategy does not gather any information about the disturbances as they occur. To reject these disturbances and account for system uncertainties, one must use alternative control strategies.

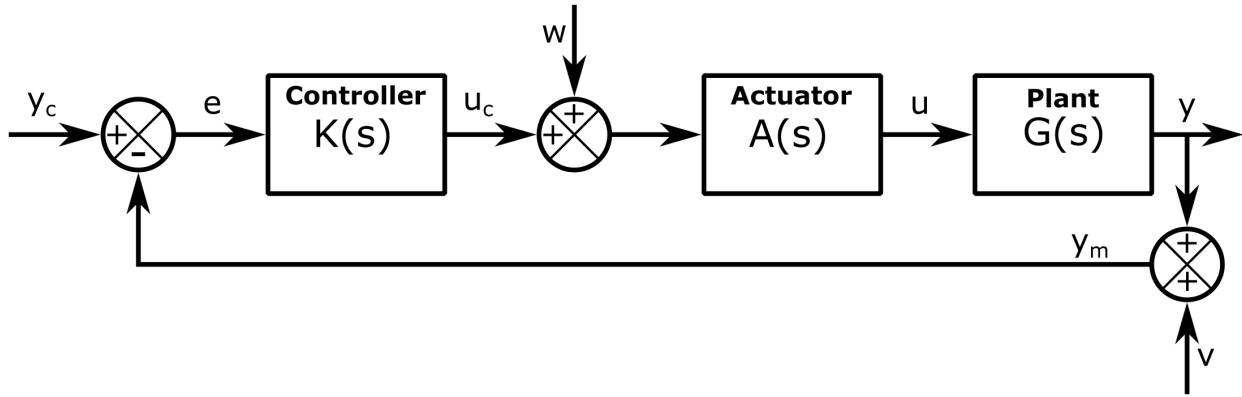
In particular, **closed-loop control** strategies, also known as **feedback control**, feed the plant output signal back to the control law, which results in a block diagram for SISO LTI systems as



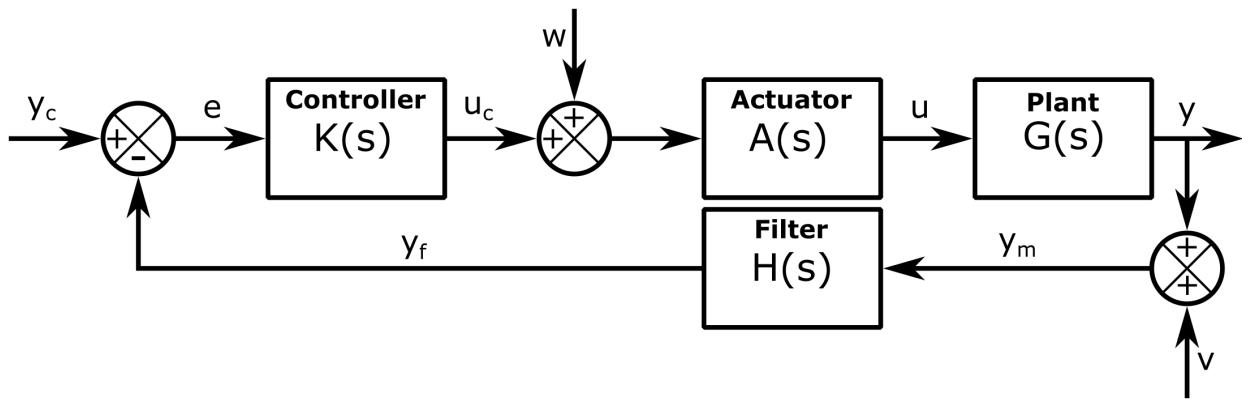
In this way, one can use an output of the plant, $y(t)$, to compute the tracking error, $e(t)$, which directly affects the control input to the plant, $u(t)$, which, in turn, affects the plant output and so on, as time continues. Recalling the negative feedback interconnection, a feedback controller produces an overall system transfer function

$$y(s) = \left[\frac{G(s)A(s)K(s)}{1 + G(s)A(s)K(s)} \right] y_c(s) \quad (2.15)$$

For real systems, the output signal must be measured by a sensor system, or **sensor**, which measures the output of the plant using physical phenomena that can be converted into a measured signal, y_m , similar to the input signal being actuated in real systems. Likewise, as no sensor provides perfect measurement, one can consider an additive noise signal, v , to the output, y , which results in a block diagram for SISO LTI systems as



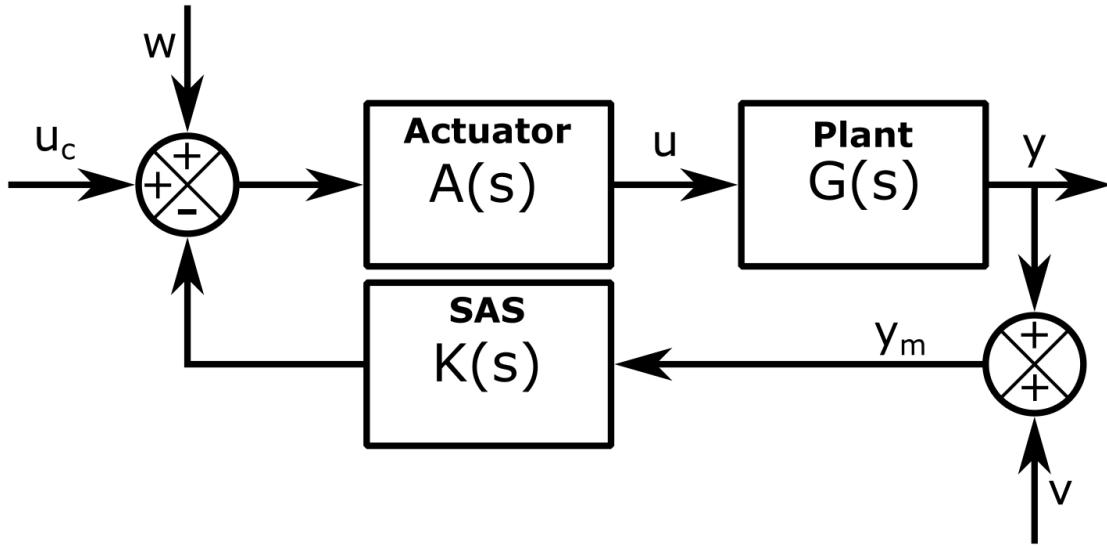
With this in mind, some control systems use a separate **filter** sub-system in the negative feedback loop to counteract the measurement noise, v , i.e.



Recalling the negative feedback interconnection, the inclusion of a separate filter into the feedback control system produces an overall system transfer function

$$y(s) = \left[\frac{G(s)A(s)K(s)}{1 + H(s)G(s)A(s)K(s)} \right] y_c(s) \quad (2.16)$$

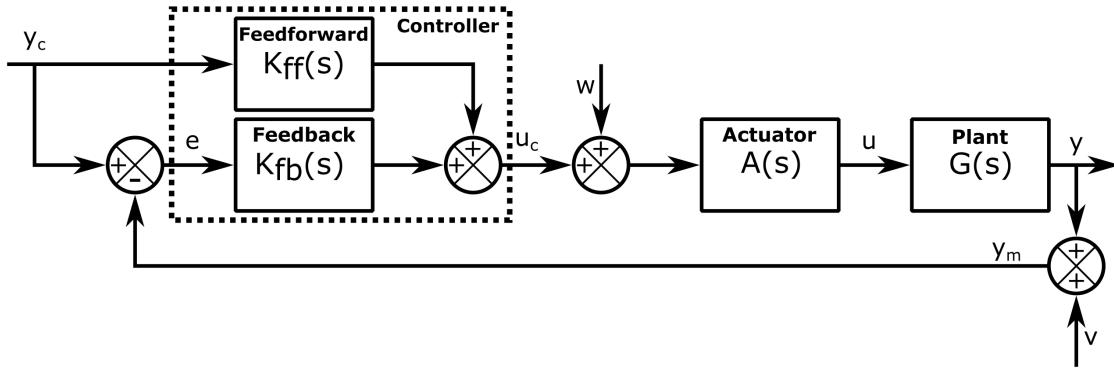
Similar to this is a special type of feedback control system called a **stability augmentation system (SAS)**. In this case, the plant is to be controlled by an external operator, e.g. pilot, but the inherent response characteristics, e.g. modal stability or damping, are not within the design specifications. Thus, a SAS is used to augment the plant dynamics to achieve certain dynamic responses, typically stability or excessive damping. An SAS for SISO LTI systems can be modeled without an actuator and sensor as the following block diagram.



Recalling the negative feedback interconnection, a SAS produces an overall system transfer function

$$y(s) = \left[\frac{G(s)A(s)}{1 + G(s)A(s)K(s)} \right] u_c(s) \quad (2.17)$$

Lastly, it should be noted that feedforward and feedback control can also be combined which results in a block diagram for SISO LTI systems as



Recalling the negative feedback and parallel interconnections, a feedback and feedforward control system produces the transfer function

$$u_c(s) = \left[\frac{K_{fb}(s)}{1 + G(s)A(s)K_{fb}(s)} + K_{ff}(s) \right] y_c(s) \quad (2.18)$$

and noting

$$y(s) = G(s)A(s)u_c(s) \quad (2.19)$$

one has an overall system transfer function

$$y(s) = \left[\frac{G(s)A(s)K_{fb}(s)}{1 + G(s)A(s)K_{fb}(s)} + G(s)A(s)K_{ff}(s) \right] y_c(s) \quad (2.20)$$

References

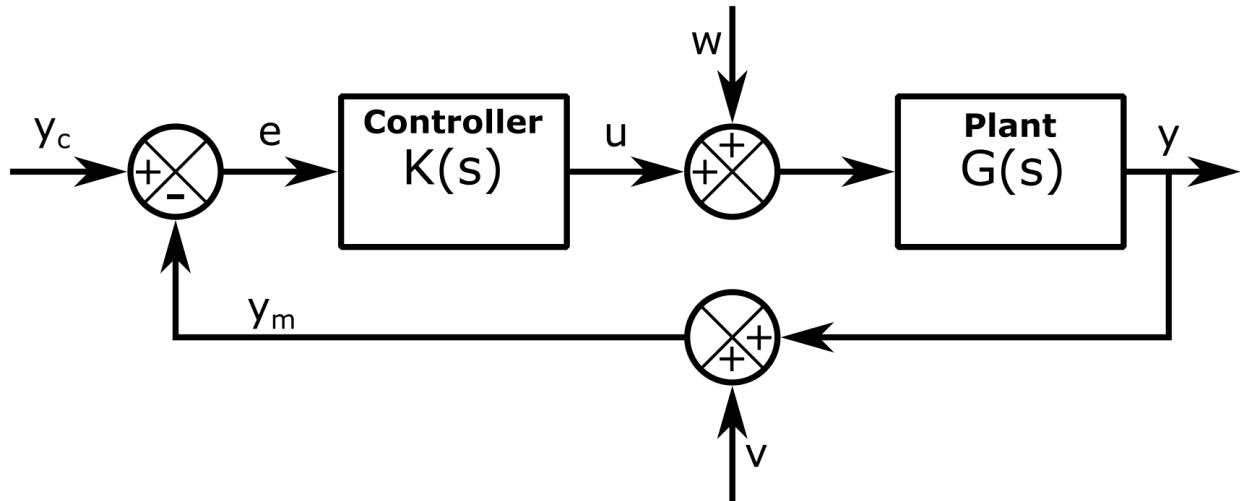
For more information, please refer to the following

- Schmidt, D. K., “12.1 Feedback Control-Law Synthesis Via Loop Shaping - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.9 Feedback Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 213-240

2.2 Classical Feedback Control System Analysis

Classical Feedback Control System

The remainder of this chapter will focus on SISO LTI control systems, the study of which is known as **classical control theory**. When designing SISO LTI feedback control systems, one usually considers the following **classical feedback control system**



where $G(s)$ is the plant (combined with potential actuator), $K(s)$ is the controller (combined with potential filter), y_c is the commanded output signal, e is the tracking error signal, u is the control input signal, w is a noise signal to the dynamic system input, e.g. disturbance, y is the system output signal, v is a noise signal on the dynamic system output, e.g. sensor noise, y_m is the measured output signal. For this system, the **open-loop transfer function** is defined as

$$L(s) = G(s)K(s) \quad (2.21)$$

and plays a large role in the analysis of the SISO LTI control system. Note that $L(s)$ would be the control system transfer function if one used an open-loop control design.

Note that this feedback control system has 3 external input signals: y_c, w, v , and 4 internal output signals: e, u, y, y_m . Then, using the previously described rules for building transfer functions, each possible input-output pair has an associated transfer function for a total of twelve and can be represented as a matrix

$$\begin{bmatrix} y(s) \\ y_m(s) \\ e(s) \\ u(s) \end{bmatrix} = \begin{bmatrix} \frac{GK}{1+GK} & \frac{G}{1+GK} & -\frac{GK}{1+GK} \\ \frac{GK}{1+GK} & \frac{G}{1+GK} & \frac{1}{1+GK} \\ \frac{1}{1+GK} & -\frac{G}{1+GK} & -\frac{1}{1+GK} \\ \frac{K}{1+GK} & -\frac{GK}{1+GK} & -\frac{K}{1+GK} \end{bmatrix} \begin{bmatrix} y_c(s) \\ w(s) \\ v(s) \end{bmatrix} \quad (2.22)$$

from which, ignoring the sign, one can see there are four **fundamental transfer functions**, namely

$$\frac{1}{1+G(s)K(s)}, \frac{G(s)K(s)}{1+G(s)K(s)}, \frac{G(s)}{1+G(s)K(s)}, \frac{K(s)}{1+G(s)K(s)} \quad (2.23)$$

Thus, selecting $K(s)$ affects these four fundamental transfer functions and the study of these transfer functions is key to classical control systems theory. Two of the four fundamental transfer functions have particular names. The first is the **error transfer function** defined as

$$S(s) = \frac{e(s)}{y_c(s)} = \frac{1}{1+G(s)K(s)} = \frac{1}{1+L(s)} \quad (2.24)$$

and is the transfer function from the closed-loop commanded output, $y_c(s)$, to the tracking error $e(s)$. The second is the **closed-loop transfer function** defined as

$$T(s) = \frac{y(s)}{y_c(s)} = \frac{G(s)K(s)}{1+G(s)K(s)} = \frac{L(s)}{1+L(s)} \quad (2.25)$$

and is the transfer function from the closed-loop commanded output, $y_c(s)$, to the actual output $y(s)$.

$S(s)$ is also known as the **sensitivity transfer function** as it characterizes how sensitive the system is to small changes in $G(s)$, an important consideration in real systems due to neglected dynamics, e.g. nonlinearities, or changing characteristics, e.g. component aging. To see this, consider the transfer function from $y_c \rightarrow y$, i.e.

$$\frac{Y(s)}{Y_c(s)} = \frac{G(s)K(s)}{1+G(s)K(s)} = T(s) \quad (2.26)$$

Now consider the ratio of a small change in T to a small change in G , i.e.

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[\frac{\Delta T}{\Delta G} \right] \left[\frac{G}{T} \right] \quad (2.27)$$

using the derivative for small Δ 's

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[\frac{dT}{dG} \right] \left[\frac{G}{T} \right] \quad (2.28)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[\frac{d}{dG} \left(\frac{GK}{1+GK} \right) \right] \left[\frac{G}{\frac{GK}{1+GK}} \right] \quad (2.29)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[\left(\frac{(1+GK)K - GK^2}{(1+GK)^2} \right) \right] \left[\frac{1+GK}{K} \right] \quad (2.30)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \left[\left(\frac{K}{(1+GK)^2} \right) \right] \left[\frac{1+GK}{K} \right] \quad (2.31)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = \frac{1}{(1+GK)} \quad (2.32)$$

$$\frac{\frac{\Delta T}{T}}{\frac{\Delta G}{G}} = S \quad (2.33)$$

Lastly, $T(s)$ is also known as the **complementary sensitivity transfer function** as one can write that

$$S(s) + T(s) = \frac{1}{1+G(s)K(s)} + \frac{G(s)K(s)}{1+G(s)K(s)} \quad (2.34)$$

or

$$S(s) + T(s) = 1 \quad \forall s \in \mathbb{C} \quad (2.35)$$

Classical Feedback Control System Stability

When designing a control system, at a minimum, one requires that the system is stable. For the classical feedback control system model, this will only occur if and only if *all* the system transfer functions are stable. However, one can simplify this criteria by considering the numerator/denominator polynomials of $G(s)$ and $K(s)$, i.e.

$$G(s) = \frac{n_G(s)}{d_G(s)} \quad (2.36)$$

$$K(s) = \frac{n_K(s)}{d_K(s)} \quad (2.37)$$

Then, the four fundamental transfer functions can be written

$$\begin{aligned} \frac{1}{1+GK} &= \frac{1}{1+\frac{n_G n_K}{d_G d_K}} = \frac{d_G d_K}{d_G d_K + n_G n_K} \\ \frac{GK}{1+GK} &= \frac{\frac{n_G n_K}{d_G d_K}}{1+\frac{n_G n_K}{d_G d_K}} = \frac{n_G n_K}{d_G d_K + n_G n_K} \\ \frac{G}{1+GK} &= \frac{\frac{n_G}{d_G}}{1+\frac{n_G n_K}{d_G d_K}} = \frac{n_G d_K}{d_G d_K + n_G n_K} \\ \frac{K}{1+GK} &= \frac{\frac{n_K}{d_K}}{1+\frac{n_G n_K}{d_G d_K}} = \frac{d_G n_K}{d_G d_K + n_G n_K} \end{aligned} \quad (2.38)$$

where notably $d_G d_K + n_G n_K$ is the denominator for each one, i.e. the **SISO feedback control system characteristic polynomial**. Thus, the **SISO feedback control system characteristic equation** is

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \quad (2.39)$$

and if the roots/poles of this equation are in the left half of the complex plane, the feedback system is stable. This is true even in the case of any pole-zero cancellations which may occur between $G(s)$ and $K(s)$. As a proof, let p_0 be a pole such that a pole/zero cancellation occurs between $G(s)$ and $K(s)$, i.e.

$$d_G(p_0) = n_K(p_0) = 0 \quad (2.40)$$

then recalling the characteristic equation

$$d_G(s)d_K(s) + n_G(s)n_K(s) = 0 \quad (2.41)$$

and substituting $s = p_0$

$$d_G(p_0)d_K(p_0) + n_G(p_0)n_K(p_0) = 0 \quad (2.42)$$

one has

$$(0)d_K(p_0) + n_G(p_0)(0) = 0 \quad (2.43)$$

which equates to

$$0 = 0 \quad (2.44)$$

In addition, note that $1 + L = 1 + GK$ appears in the denominator of each fundamental transfer function which can be rewritten as

$$1 + L(s) = 1 + G(s)K(s) = 1 + \frac{n_G n_K}{d_G d_K} = \frac{d_G(s)d_K(s) + n_G(s)n_K(s)}{d_G(s)d_K(s)} \quad (2.45)$$

Thus, one can see that the numerator of $1+L(s)$ is the SISO feedback control system characteristic polynomial. However, as $1 + L(s)$ can still be affected by pole-zero cancellations, one can equivalently declare that one has a **stable SISO feedback control system** if and only if

1. there are no RHP pole-zero cancellations when forming $L(s)$; and
2. $1 + L(s)$ has no zeros in the RHP.

The analysis of the zeros of $L(s)$ as regards stability and robustness will be discussed in the following lectures.

References

For more information, please refer to the following

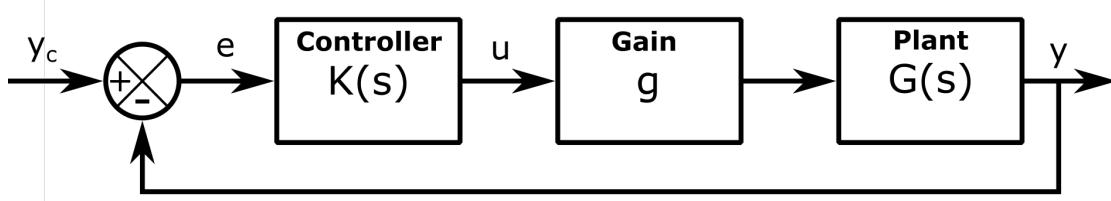
- Schmidt, D. K., “12.1 Feedback Control-Law Synthesis Via Loop Shaping - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.9 Feedback Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 213-240

2.3 Classical Feedback Control System Stability Margins

As mentioned previously, the plant model in a feedback control system is typically only an approximation due to model uncertainty, e.g. linearization error, model parameter variability, simplified dynamics modeling, etc. This is especially true for FDC. To account for this, control engineers use safety factors when designing feedback control systems. These safety factors are known as **stability margins** and quantify the **feedback control system robustness**, i.e. a measure of the feedback control system will also perform adequately well under a different set of assumptions for the system model. This section will consider four primary SISO stability margins: gain margin, phase margin, delay margin, and disk margin.

Gain Margin

The **gain margin** is the amount of gain on the nominal plant model that can be added before closed-loop instability. The gain margin can be derived by considering a scaling perturbation to the plant dynamics model, $G(s)$, which can be modeled in block diagram form as



where $g > 0$ is some scalar constant. Then, defining the perturbed open-loop transfer function as $gL(s)$, the gain margin is determined by assessing for what values of g the feedback control system will go unstable.

Recalling that the feedback control system is stable if (1) there are no pole-zero cancellations and (2) the zeros of $1 + L(s) = 0$ are only in the LHP. Then, a critical gains, g_0 , occurs when the zeros of $1 + g_0L(j\omega_0)$ are on the imaginary axis for some critical frequency ω_0 , i.e.

$$1 + g_0L(j\omega_0) = 0 \quad (2.46)$$

Rewriting this expression, one has

$$L(j\omega_0) = -\frac{1}{g_0} \quad (2.47)$$

or in polar form (i.e. gain and phase), one has

$$|L(j\omega_0)|e^{j\angle L(j\omega_0)} = \frac{1}{g_0}e^{\pm j\pi} \quad (2.48)$$

or the equivalent magnitude/gain and phase requirements become

$$g_0 = \frac{1}{|L(j\omega_0)|} \quad (2.49)$$

and

$$\angle L(j\omega_0) = \pm 180^\circ \quad (2.50)$$

Thus, one can identify the gain margins by:

1. identifying all critical frequencies $\omega_{0,i}$ where $\angle L(j\omega_{0,i}) = \pm 180^\circ$;
2. calculating all associated “candidate” gain margins, $g_{0,i}$, using the inverse of the gain at $\omega_{0,i}$, i.e.

$$g_{0,i} = \frac{1}{|L(j\omega_{0,i})|} \quad (2.51)$$

3. setting the **upper gain margin**, \bar{g} , as $\min_i g_{0,i} > 1$; and
4. setting the **lower gain margin**, \underline{g} , as $\max_i g_{0,i} < 1$.

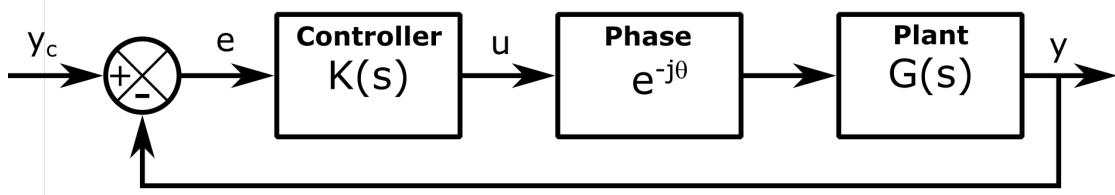
which can be done simply from the Bode plot of $L(j\omega)$. Then, one may state that the feedback control system is stable for

$$\underline{g} \leq g \leq \bar{g} \quad (2.52)$$

where it should be noted that not all systems will have an upper *and* lower gain margin. For FDC, a feedback control system is generally considered sufficiently robust if $\underline{g} \leq 0.5$ and $\bar{g} \geq 2$, i.e. $\underline{g} \leq -6$ dB and $\bar{g} \geq 6$ dB, for the gain margin.

Phase and Delay Margin

The **phase margin** is the amount of phase lead/lag that can be added to the input before closed-loop instability. The phase margin can be derived by considering a phase perturbation to the plant dynamics model which can be modeled in block diagram form as



Then, defining the perturbed open-loop transfer function as $e^{-j\theta}L(s)$ the phase margin is determined by assessing for what values of θ the feedback control system will go unstable.

Recalling that the feedback control system is stable if (1) there are no pole-zero cancellations and (2) the zeros of $1 + L(s) = 0$ are only in the LHP. Then, a critical phase, θ_0 , occurs when the zeros of $1 + e^{-j\theta}L(j\omega_0)$ are on the imaginary axis for some critical frequency ω_0 , i.e.

$$1 + e^{-j\theta_0}L(j\omega_0) = 0 \quad (2.53)$$

Rewriting this expression, one has

$$L(j\omega_0) = -e^{j\theta_0} \quad (2.54)$$

or in polar form (i.e. gain and phase), one has

$$|L(\omega_0)|e^{j\angle L(\omega_0)} = e^{\pm j\pi}e^{j\theta_0} = e^{j(\pm\pi+\theta_0)} \quad (2.55)$$

or the equivalent magnitude/gain and phase requirements become

$$|L(j\omega_0)| = 1 \quad (2.56)$$

and

$$\theta_0 = \pm 180^\circ + \angle L(j\omega_0) \quad (2.57)$$

where θ_0 is typically represented as a positive number as the phase margin can be referenced to either $\pm 180^\circ$ and is symmetric for positive and negative θ .

Thus, one can identify the phase margin by:

1. identifying critical frequencies $\omega_{0,i}$ where $|L(j\omega_{0,i})| = 1 = 0 \text{ dB}$;
2. calculating associated “candidate” phase margins, $\theta_{0,i}$, using distance from 180° , i.e.

$$\theta_{0,i} = \pm 180^\circ + \angle L(j\omega_0) \quad (2.58)$$

3. setting the phase margin, $\bar{\theta}$, as $\min_i |\theta_{0,i}|$

which can be done simply from the Bode plot of $L(j\omega)$. Then, one may state that the feedback control system is stable for

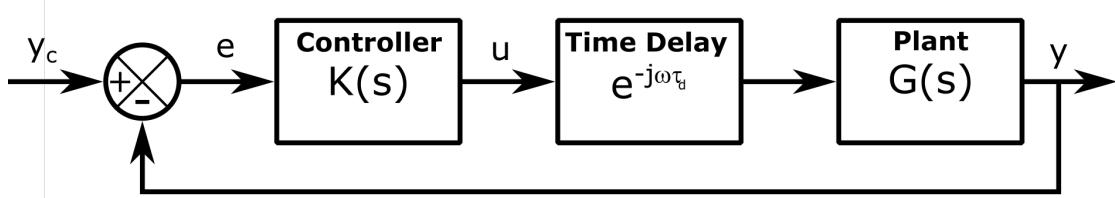
$$-\bar{\theta} \leq \theta \leq \bar{\theta} \quad (2.59)$$

For FDC, a feedback control system is generally considered sufficiently robust if $\bar{\theta} \geq 45^\circ$ for the phase margin.

Alternatively, the phase margin can be rewritten in terms of a (time) delay margin. The **(time) delay margin** is the amount of time delay that can be added to the input before closed-loop instability. To which uses the fact that the critical time delay, τ_d , is related to θ_0 by

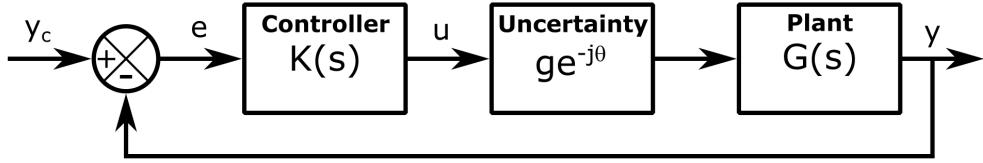
$$\tau_d = \frac{\theta_0}{\omega_0} \quad (2.60)$$

which can also be modeled in block diagram form as

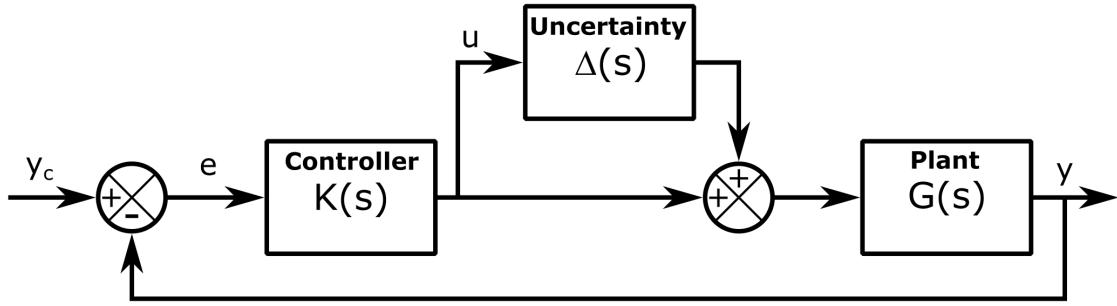


Disk Margin

Consider a simultaneous gain and phase uncertainty model, i.e.



which encodes an arbitrary frequency response uncertainty to the nominal feedback control system. Alternatively, one could reform this form of uncertainty as $\Delta(s)$ which can be modeled in block diagram form as



where $\Delta(s) = 0$ refers to the nominal model.

Recalling that the feedback control system is stable if (1) there are no pole-zero cancellations and (2) the zeros of $1 + L(s) = 0$ are only in the LHP. Then, a critical uncertainty, $\Delta(j\omega_0)$, occurs when the zeros of $1 + (1 + \Delta(j\omega_0)) L(j\omega_0)$ are on the imaginary axis for some critical frequency ω_0 , i.e.

$$1 + (1 + \Delta(j\omega_0)) L(j\omega_0) = 0 \quad (2.61)$$

Rewriting this expression, one has

$$1 + L(j\omega_0) = \Delta(j\omega_0) - \Delta(j\omega_0)L(j\omega_0) \quad (2.62)$$

or in polar form (i.e. gain and phase), one has

$$\|1 + L(j\omega_0)\|e^{-j\angle(1+L(j\omega_0))} = \|\Delta(j\omega_0)L(j\omega_0)\|e^{j(\pm\pi-\angle(\Delta(j\omega_0)L(j\omega_0)))} \quad (2.63)$$

or the equivalent magnitude/gain and phase requirements become

$$\|\Delta(j\omega_0)L(j\omega_0)\| = \|1 + L(j\omega_0)\| \quad (2.64)$$

and

$$\angle(\Delta(j\omega_0)L(j\omega_0)) = \pm 180^\circ + \angle(1 + L(j\omega_0)) \quad (2.65)$$

Then, one can define the **disk margin**, d_{min} , as the minimum amount $\Delta(j\omega_0)L(j\omega_0)$ adds to $L(j\omega_0)$ at some ω_0 to create instability, i.e. the minimum perturbation “distance” allowable before closed-loop instability

$$d_{min} = \min_{0 \leq \omega < \infty} \|\Delta(j\omega_0)L(j\omega_0)\| \quad (2.66)$$

which, by the equivalence above, can also be considered as the minimum distance between $L(j\omega)$ and the $-1 \pm 0j$ point in the complex plane

$$d_{min} = \min_{0 \leq \omega < \infty} |1 + L(j\omega)| \quad (2.67)$$

which is minimized the closer $L(j\omega)$ is to $-1 \pm 0j$. This also can be written in terms of the sensitivity transfer function as

$$d_{min} = \max_{0 \leq \omega < \infty} |S(j\omega)| \quad (2.68)$$

as $|S(j\omega)| = |1 + L(j\omega)|^{-1}$.

Furthermore, the gain and phase margins are two different “directions” in the complex domain that one is assessing the distance from $-1 \pm 0j$ point. In particular, the gain margin criterion in polar form (i.e. gain and phase) is

$$\angle L(j\omega_0) = \pm 180^\circ \quad (2.69)$$

which restricts the minimization of the distance of $L(j\omega)$ from -1 along the negative real axis for the gain margin, i.e.

$$d_{min,gm} = \min_{0 \leq \omega < \infty, \angle L(j\omega_0) = \pm 180^\circ} |1 + L(j\omega)| \quad (2.70)$$

or, as the constraint, $\angle L(j\omega_0) = \pm 180^\circ$ means $L(j\omega)$ is a negative real number, one has

$$d_{min,gm} = \begin{cases} \min_{0 \leq \omega < \infty, \angle L(j\omega_0) = \pm 180^\circ} 1 - |L(j\omega)| & L(j\omega) < -1 \\ \min_{0 \leq \omega < \infty, \angle L(j\omega_0) = \pm 180^\circ} |L(j\omega)| - 1 & L(j\omega) > -1 \end{cases} \quad (2.71)$$

and substituting $|L(j\omega_0)| = \frac{1}{g_0}$, one has

$$d_{min,gm} = \begin{cases} 1 - \frac{1}{g} = \min_{0 \leq \omega < \infty, \angle L(j\omega_0) = \pm 180^\circ} 1 - |L(j\omega)| & -1 < L(j\omega) < 0 \\ \frac{1}{g} - 1 = \min_{0 \leq \omega < \infty, \angle L(j\omega_0) = \pm 180^\circ} |L(j\omega)| - 1 & L(j\omega) < -1 \end{cases} \quad (2.72)$$

Furthermore, the phase margin criterion in polar form (i.e. gain and phase) is

$$|L(j\omega_0)| = 1 \quad (2.73)$$

which restricts the minimization to the distance of $L(j\omega)$ from -1 along the unit circle for the phase margin, i.e.

$$d_{min,pm} = \min_{0 \leq \omega < \infty, |L(j\omega_0)| = 1} |1 + L(j\omega)| \quad (2.74)$$

or

$$d_{min,pm} = \min_{0 \leq \omega < \infty, |L(j\omega_0)| = 1} |1 + e^{j\angle L(j\omega_0)}| \quad (2.75)$$

and substituting $\angle L(j\omega_0) = \pm\pi + \theta_0$, one has

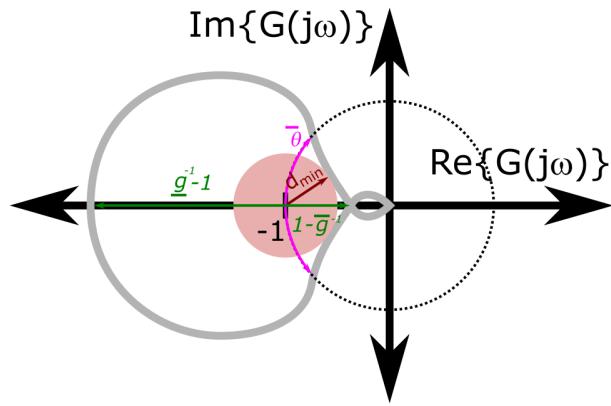
$$d_{min,pm} = \begin{cases} |1 + e^{j(\bar{\theta} + \pi)}| = \min_{0 \leq \omega < \infty, |L(j\omega_0)| = 1} |1 + e^{j\angle L(j\omega_0)}| & -\pi < \angle L(j\omega_0) < 0 \\ |1 + e^{j(\bar{\theta} - \pi)}| = \min_{0 \leq \omega < \infty, |L(j\omega_0)| = 1} |1 + e^{j\angle L(j\omega_0)}| & 0 < \angle L(j\omega_0) < \pi \end{cases} \quad (2.76)$$

Furthermore, the phase margin assesses the arc distance from $-1 \pm 0j$ along the unit circle, i.e.

$$\theta_0 = \pm\pi + \angle\Delta(j\omega_0)L(j\omega_0) \quad (2.77)$$

With these definitions, a disk margin of 0.4 corresponds to a gain margins of $\underline{g} = 0.71$ and $\bar{g} = 1.67$ and a phase margin of $\bar{\theta} = 23^\circ$. However, the disk margin can also account for simultaneous gain and phase perturbations to the nominal feedback control system.

These relationships can be represented graphically in the Nyquist plot as



where the term “disk” derives from the disk that one forms this minimum distance that just intersects the frequency response of $L(j\omega)$.

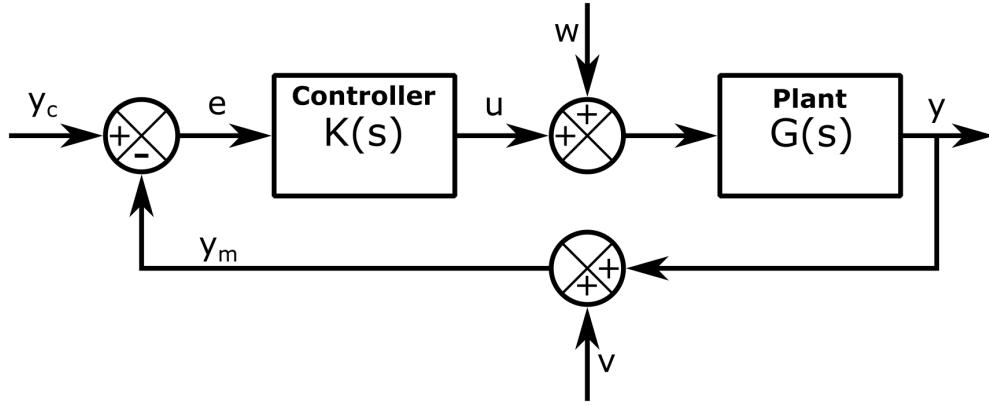
References

For more information, please refer to the following

- Schmidt, D. K., “12.1 Feedback Control-Law Synthesis Via Loop Shaping - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.9 Feedback Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 213-240

2.4 SISO Loop-Shaping Control Design Requirements

Recall the general feedback control system



with the objective to synthesize a controller $K(s)$ such that it generally satisfies the following qualitative design requirements:

- 1 stable with good stability margins;
- 2 good tracking, i.e. the transfer function from $y_c \rightarrow e$ is small;
- 3 disturbance rejection, i.e. the transfer function from $w \rightarrow y$ is small;
- 4 sensor noise filtering, i.e. the transfer function from $v \rightarrow e$ is small; and
- 5 the control effort is realistic, i.e. $|u|$ is not too large.

To satisfy all stability and performance requirements for SISO systems , one can use **loop-shaping** control design methods for synthesizing $K(s)$ through “shaping” the SISO open-loop transfer function, $L(s) = G(s)K(s)$, to have certain frequency response characteristics. This section will establish the connections between these qualitative design requirements 1-5 and stability and performance requirements on $L(j\omega)$.

The primary aspect of loop-shaping is that $K(s)$ has an additive effect on the magnitude subplot of the Bode plot when considered in decibels, i.e.

$$20 \log_{10} |L(j\omega)| = 20 \log_{10} |G(j\omega)K(j\omega)| = 20 \log_{10} |G(j\omega)| + 20 \log_{10} |K(j\omega)| \quad (2.78)$$

Thus, to form a suitable $L(s)$, one can use multiple **control stages** for different regions of the Bode plot using this additive property for shaping $L(s)$. Then, multiplying each stage together, i.e. using each stage in *series*, will provide the full controller transfer function $K(s)$, i.e. for N stages,

$$20 \log_{10} |K_1(j\omega) \cdots K_N(j\omega)| = 20 \log_{10} |K_1(j\omega)| + \cdots + 20 \log_{10} |K_N(j\omega)| \quad (2.79)$$

Open-Loop Transfer Function Stability Requirements

Design requirement 1 specifies the feedback control system must be stable, i.e. the roots of the characteristic polynomial must be in the LHP, typically checked through the zeros of $L(s)$ as there should also not be any pole-zero cancellations. In addition, one typically also requires some stability margin for the system due to uncertainty in the plant model $G(s)$. To provide some guidance on connecting stability margins to

the loop-shaping control design method consider the following derivation. The **Bode gain-phase formula** states that if $L(0) > 0$ and has all its poles and zeros in LHP, then the phase of $L(j\omega)$ (in radians) at ω is

$$\angle L(j\omega) = \frac{1}{\pi} \int_{-\infty}^{\infty} \frac{d|L|_{\text{dB}}(\zeta)}{d\nu} \log_{10} \coth \left| \frac{\nu}{2} \right| d\nu \quad (2.80)$$

where $\nu = \log_{10} \frac{\zeta}{\omega}$. Next, noting that

$$\log_{10} \coth \left| \frac{\nu}{2} \right| \approx \frac{1}{2} \pi^2 \delta(\nu) \quad (2.81)$$

one has the approximation

$$\angle L(j\omega) \approx \frac{\pi}{2} \left(\frac{d|L|_{\text{dB}}(\zeta)}{d\nu} \right)_{\zeta=\omega} \quad (2.82)$$

or in degrees

$$\angle L(j\omega) \approx \frac{90^\circ}{20 \text{ dB/decade}} (|L(j\omega)|_{\text{dB}}/\text{decade}) \quad (2.83)$$

Thus, if $|L(j\omega)|_{\text{dB}}/\text{decade} = -30 \text{ dB/decade}$ over roughly 1 decade of ω , then the phase approximately satisfies

$$\angle L(j\omega) \approx \frac{90^\circ}{20 \text{ dB/decade}} (-30 \text{ dB/decade}) \quad (2.84)$$

or

$$\angle L(j\omega) \approx -135^\circ \quad (2.85)$$

Then, recall that one typically requires a phase margin requirement of at least $\pm 45^\circ$ and a gain margin requirement of at least $\pm 6 \text{ dB}$, which occur at $\angle L(j\omega) = \pm 180^\circ$, the open-loop transfer function, $L(s)$, should roughly satisfy the following:

1. no poles or zeros in the RHP;
2. $|L(0)| > 0$;
3. a single gain crossover frequency ω_c ;
4. $|L(j\omega)|_{\text{dB}}/\text{decade} \geq -30 \text{ dB/decade}$ for $\frac{\omega_c}{\sqrt{10}} < \omega < \sqrt{10}\omega_c$;
5. $|L|_{\text{dB}} \geq 6 \text{ dB}$ for $\omega \leq \frac{\omega_c}{\sqrt{10}}$; and
6. $|L|_{\text{dB}} \leq -6 \text{ dB}$ for $\omega \geq \sqrt{10}\omega_c$.

Then, one can confidently claim that the feedback control system achieves the stability requirements.

Though these assumptions do not hold for all LTI systems, this approximation is generally decent and provides at least a good starting point for how much to limit the gain slope about ω_c . It should be noted that if $L(s)$ has zeros or poles in the RHP, then there may not exist a suitable $K(s)$, but at the very least there will be some additional phase introduced at ω_c which will require a larger slope than -30 dB/decade . This additional phase is why a system with no poles or zeros in the RHP is called a **minimum phase system**. For non-minimum phase systems, additional control design must be performed case-by-case. Lastly, it should be noted that the *single* crossover frequency ω_c for $|L(j\omega_c)| = 0 \text{ dB}$ is also known as the **loop bandwidth** for the feedback control system, and is typically specified for some required frequency level in the design requirements.

Open-Loop Transfer Function Performance Requirements

Design requirement 2 specifies the feedback control system to have good tracking, i.e. keep e small. Recalling the previously derived transfer function for $y_c \rightarrow e$, i.e.

$$\frac{e(s)}{y_c(s)} = \frac{1}{1 + GK} = \frac{1}{1 + L} = S \quad (2.86)$$

one requires that $|\frac{1}{1+L}| \ll 1$ which occurs if $|L| \gg 1$. Furthermore, note that

$$|S(s)| = \left| \frac{1}{1 + L(s)} \right| \quad (2.87)$$

$$|1 + L(s)| = \left| \frac{1}{S(s)} \right| \quad (2.88)$$

and the triangle inequality, i.e.

$$|a + b| \leq |a| + |b| \quad (2.89)$$

Then, one can state

$$|1 - 1 + L(s)| \leq |-1| + |1 + L(s)| \quad (2.90)$$

$$|L(s)| \leq 1 + |1 + L(s)| \quad (2.91)$$

$$|L(s)| - 1 \leq |1 + L(s)| \quad (2.92)$$

Thus, for a requirement on $|S(j\omega)| \leq e_{\omega,req}$, one has that

$$|e| \leq e_{\omega,req} \rightarrow |\frac{1}{S(j\omega)}| \geq \frac{1}{e_{\omega,req}} \quad (2.93)$$

$$|e| \leq e_{\omega,req} \rightarrow |1 + L(j\omega)| \geq \frac{1}{e_{\omega,req}} \quad (2.94)$$

or, using the inequality above, one can define the approximate relationship

$$|e| \leq e_{\omega,req} \rightarrow |L(j\omega)| \geq 1 + \frac{1}{e_{\omega,req}} \quad (2.95)$$

Design requirement 3 specifies the feedback control system to reject disturbances, i.e. w should have little effect on y . Recalling the previously derived transfer function for $w \rightarrow y$, i.e.

$$\frac{y(s)}{w(s)} = \frac{G}{1 + GK} = \frac{G}{1 + L} \quad (2.96)$$

one requires that $|\frac{G}{1+L}| \ll 1$. However, noting that the effect of the disturbance with feedback control should be at least smaller than with no control, i.e. $|\frac{G}{1+L}| \ll |G|$ or $|\frac{1}{1+L}| \ll 1$, which occurs if that $|L| \gg 1$.

Design requirement 4 specifies the feedback control system to filter out the sensor noise, i.e. v should have little effect on e . Recalling the previously derived transfer function for $v \rightarrow e$, i.e.

$$\frac{e(s)}{v(s)} = \frac{-GK}{1 + GK} = \frac{-L}{1 + L} \quad (2.97)$$

one requires that $|\frac{L}{1+L}| \ll 1$ which occurs if $|L| \ll 1$. For example, one typically requires filtering out noise at high frequencies, i.e.

$$|T(j\omega)| \leq g \quad \forall \omega > \omega_{high} \quad (2.98)$$

But by definition

$$T(j\omega) = \frac{L(j\omega)}{1 + L(j\omega)} \quad (2.99)$$

Thus, one requires that

$$\left| \frac{1 + L(j\omega)}{L(j\omega)} \right| \leq \frac{1}{g} \quad (2.100)$$

or if

$$\left| \frac{1}{L(j\omega)} \right| \geq \frac{1}{g} + 1 \quad (2.101)$$

then, by the triangle inequality,

$$\left| \frac{1 + L(j\omega)}{L(j\omega)} \right| \geq \left| \frac{1}{L(j\omega)} \right| - 1 \geq \frac{1}{g} \quad (2.102)$$

Then, these type of requirements can be converted to

$$\left| \frac{1}{L(j\omega)} \right| \leq \frac{1+g}{g} \quad (2.103)$$

or

$$|L(j\omega)| \leq \frac{g}{1+g} \quad \forall \omega > \omega_{high} \quad (2.104)$$

Design requirement 5 specifies the feedback control system to keep u small enough. Recalling the previously derived transfer function for $y_c \rightarrow u$, i.e.

$$\frac{u(s)}{y_c(s)} = \frac{K}{1 + GK} \quad (2.105)$$

one requires that $|\frac{K}{1+L}| \ll 1$. Because G is fixed, one has 2 cases:

1. If $|G| \gg 1$, then all $|K|$ will provide $|\frac{K}{1+GK}| \ll 1$
2. If $|G| \ll 1$, then $|K| \ll 1$ will provide $|\frac{K}{1+GK}| \ll 1$

Summarizing these rough guidelines results in the following table.

| General Requirement | Closed-Loop TF Requirement | Loop TF Requirement |
|--------------------------|----------------------------|-----------------------------------|
| 1. Good Tracking | $ S \ll 1$ | $ L \gg 1$ |
| 2. Disturbance Rejection | $ S \ll 1$ | $ L \gg 1$ |
| 3. Noise Filtering | $ T \ll 1$ | $ L \ll 1$ |
| 4. Small control effort | $ KS \ll 1$ | If $ G \ll 1$, then $ K \ll 1$ |

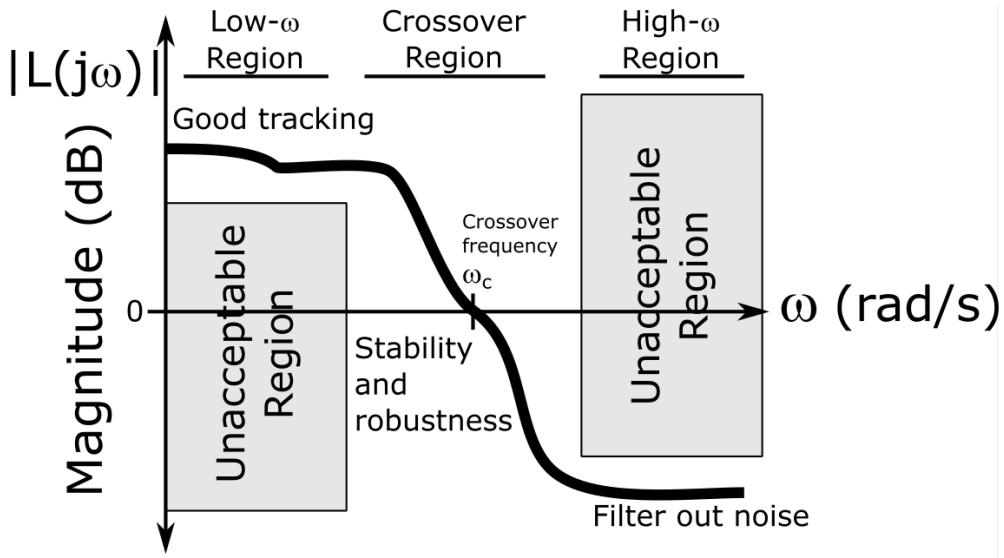
Thus, the stipulation that $S+T = 1$ demonstrates that one cannot simultaneously satisfy all control design requirements. Thus, the *key idea* in frequency domain control design is that output commands are designed to be low frequency signals relative to any high frequency noise signals on the actual output. Thus, one requires the feedback control system at least satisfies

1. $|S(j\omega)| \ll 1$ at low ω
2. $|T(j\omega)| \ll 1$ at high ω

Then, translating these to the open-loop transfer function, $L(j\omega)$, provides the equivalent loop-shaping performance requirements

1. $|L(j\omega)| \gg 1$ at low ω
2. $|L(j\omega)|_{\text{dB}}/\text{decade} \geq -30 \text{ dB/decade}$ around ω_c
3. $|L(j\omega)| \ll 1$ at high ω

which can be analyzed visually using the magnitude subplot of the Bode plot of $L(j\omega)$, i.e.



For the control effort requirement, one must analyze the Bode plot of $K(j\omega)S(j\omega)$ in parallel with the loop-shaping of $L(s)$ in these three regions. As a general rule, one typically should design $K(j\omega)$ to not be too large where $G(j\omega)$ is small. This is similar to the high-frequency requirement for sensor noise filtering and suppression of high-frequency model uncertainty.

Frequency and Step Response Characteristics Relationships

Alternatively, in some cases, the design requirements for a control law may be specified as certain unit step response characteristics, i.e.

$$6 \quad t_s \leq t_{s,req}$$

$$7 \quad M_p < M_{p,req}$$

$$8 \quad |e_{ss}| \leq e_{ss,req}$$

where $t_{s,req}$, $e_{ss,req}$, and u_{req} are some values where $e_{ss,req}$ and u_{req} may alternatively be percentages. This subsection will discuss these alternative step response requirements and their loose relationships to the loop-shaping design requirements.

For design requirement 6, $t_s \leq t_{s,req}$, as a rough approximation, the loop bandwidth is related to the settling time roughly by

$$t_s \approx \frac{3}{\omega_c} \quad (2.106)$$

Thus, one can define the approximate relationship that

$$t_s \leq t_{s,req} \rightarrow \omega_c \text{ of } L(j\omega) = 0 \text{ dB} \geq \frac{3}{t_{s,req}} \quad (2.107)$$

However, for some systems this approximation can be significantly off, but, in general, increasing ω_c will decrease t_s and is done through design iteration.

For design requirement 7, $M_p < M_{p,req}$, as a rough approximation, the phase margin can be related to the damping ratio roughly by

$$\bar{\theta} \approx 100\zeta \quad (2.108)$$

and using the second-order expression for the maximum overshoot in terms of ζ , i.e.

$$M_p \approx e^{\frac{\zeta\pi}{\sqrt{1-\zeta^2}}} \quad (2.109)$$

One can define the approximate relationship that

$$M_p < M_{p,req} \rightarrow \bar{\theta} > 100 \sqrt{\frac{\ln M_{p,req}^2}{\pi^2 + \ln M_{p,req}^2}} \quad (2.110)$$

However, for some systems this approximation can be significantly off, but, in general, increasing $\bar{\theta}$ will decrease M_p and is done through design iteration.

For design requirement 8, $|e_{ss}| \leq e_{ss,req}$ can be regarded as a requirement on $|S(0)| \leq e_{ss,req}$, as $\omega = 0$ describes the step input. Thus, one can define the exact relationship that

$$|e_{ss}| \leq e_{ss,req} \rightarrow |L(0)| \geq 1 + \frac{1}{e_{ss,req}} \quad (2.111)$$

which is simply another form of the low frequency requirement on $L(j\omega)$.

References

For more information, please refer to the following

- Nelson, R. C., “7.5 Time-Domain and Frequency-Domain Specifications,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 251-262
- Schmidt, D. K., “12.1 Feedback Control-Law Synthesis Via Loop Shaping - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.9 Feedback Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 213-240

2.5 Fundamental SISO Loop-Shaping Control Stages

Loop-shaping uses multiple **N control stages** in order to shape different frequency regions of $L(s)$. Thus, one designs controllers in *series form* as

$$K(s) = K_1(s) \cdots K_N(s) \quad (2.112)$$

which is notably additive in the Bode plot which serves as the primary analysis tool. As each stage is in series, multiplying each stage together provides the controller transfer function $K(s)$. The four most common control stages are:

1. Proportional
2. Integral
3. Derivative
4. Low-Pass Filter

while four others are also common in some applications

1. High-Pass Filter
2. Band-Pass Filter
3. Low-Frequency Boost
4. Lead-Lag

It should be noted that all control stages are not necessary for a sufficient controller and typically the least number of stages, i.e., less complexity, is preferred by control engineers.

With this in mind, for the general SISO loop-shaping control design, one can consider the following iterative procedure for **loop-shaping design** of $K(s)$ until all design requirements are satisfied.

1. Use a proportional control stage to set the loop bandwidth, ω_c
 - May need to obtain requirements on ω_c from t_s
2. Use integral control stage(s) to increase $|L(j\omega)|$ at low ω , i.e. good tracking.
 - May need to obtain requirements on $L(j\omega)$ from $S(j\omega)$ or e_{ss}
3. Use derivative control stage(s) to reduce $|L(j\omega)|$ slope about ω_c to $> -30 \text{ dB}/10\omega$ for stability and robustness.
 - Will need to lower proportional gain based on new loop bandwidth
4. Use low-pass filter control stage(s) to decrease $|L(j\omega)|$ at high ω , i.e. good noise filtering.
 - May need to obtain requirements on $L(j\omega)$ from $T(j\omega)$

It should be noted that iteration must typically be used as the control stages may slightly affect other frequency regions.

Proportional Control Stage

This control stage is used to shift $L(s)$ up or down, i.e. to set the crossover frequency, ω_c , a.k.a. the bandwidth, in the crossover-frequency region to have the system respond as fast as necessary. The transfer function for this control stage is defined as

$$K(s) = K_p \quad (2.113)$$

where K_p is the gain. Note that $|K_p| > 1$ shifts $L(s)$ up while $|K_p| < 1$ shifts $L(s)$ down. Note also that $K_p > 0$ or $K_p < 0$ should be chosen so that the feedback control system is stabilized while also not being too large as to violate any control effort constraints.

As an example, consider the following system

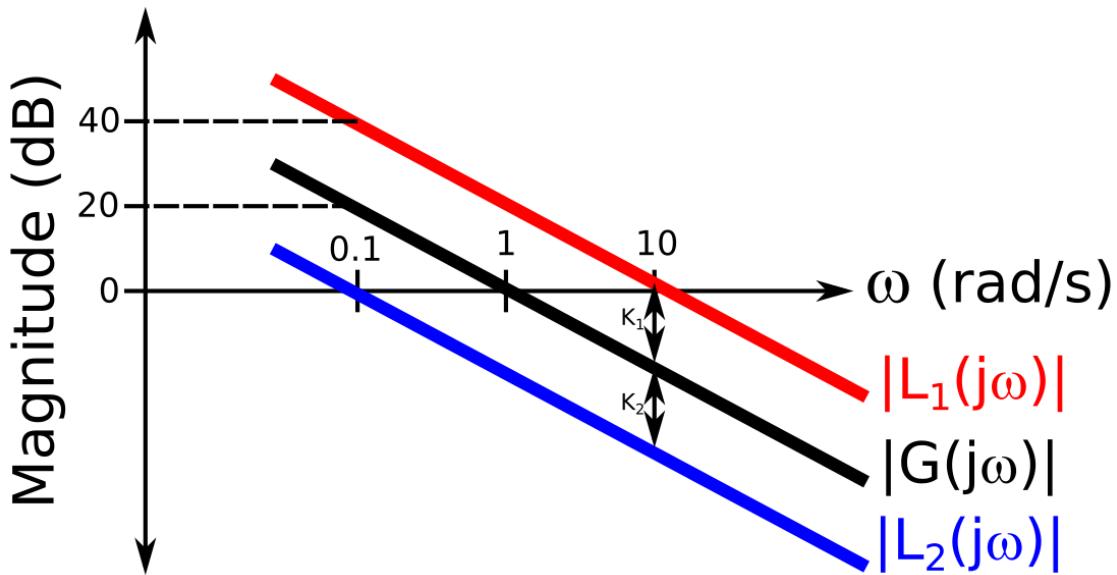
$$G(s) = \frac{1}{s} \quad (2.114)$$

$$K_1(s) = 10 \quad (2.115)$$

$$K_2(s) = 0.1 \quad (2.116)$$

$$L_1(s) = \frac{10}{s} \quad (2.117)$$

$$L_2(s) = \frac{0.1}{s} \quad (2.118)$$



Integral Control Stage

This control stage is used to increase the gain of $L(s)$ for the low-frequency region $\omega < \omega_i$, i.e. to have good tracking at low frequencies. The transfer function for this control stage is defined as

$$K(s) = \frac{s + \omega_i}{s} \quad (2.119)$$

where ω_i is the chosen corner frequency/bandwidth to start the gain increase. This stage has the following notable properties:

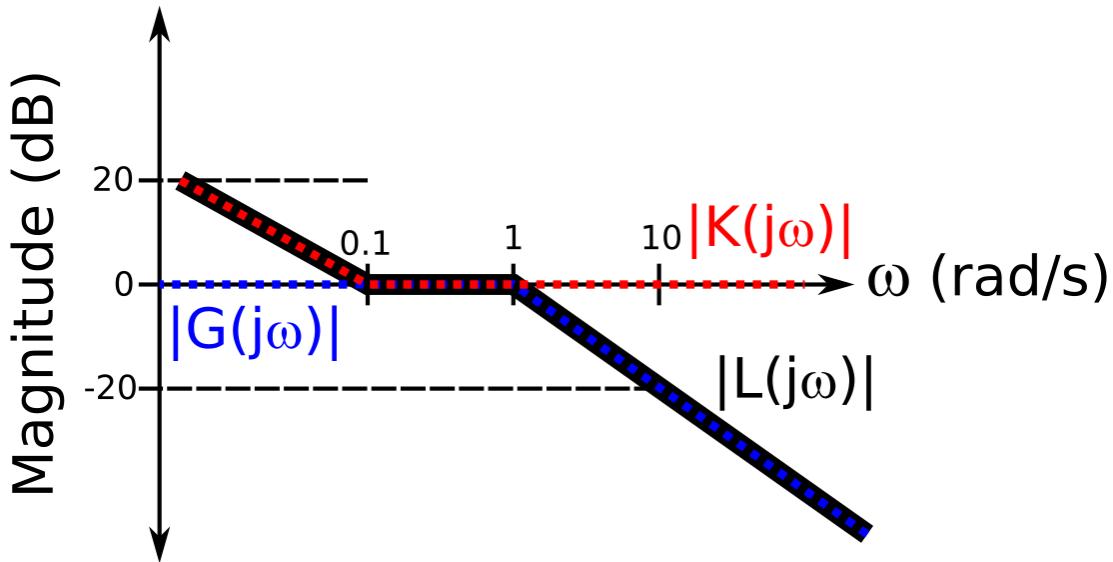
1. $K(s)$ has a pole at $s = 0$ and a zero at $-\omega_i$;
2. $K(j\omega) \rightarrow +\infty$ as $\omega \rightarrow 0$, thus $e_{ss} = 0$;
3. at low ω , $K(j\omega) \approx \frac{\omega_i}{j\omega}$, i.e. for every $10\omega \rightarrow K(j\omega)/10$ or a -20 dB/10ω slope; and
4. at high ω , $K(j\omega) \approx 1$, i.e. no effect (0 dB).

As an example, consider the following system

$$G(s) = \frac{1}{s + 1} \quad (2.120)$$

$$K(s) = \frac{s + 0.1}{s} \quad (\omega_i = 0.1) \quad (2.121)$$

$$L(s) = \frac{s + 0.1}{s^2 + s} \quad (2.122)$$



Derivative Control Stage

This control stage is used to increase the slope of $L(j\omega)$ for the crossover-frequency region $\omega > \omega_d$. The transfer function for this control stage is defined as

$$K(s) = \frac{1}{\omega_d} s + 1 \quad (2.123)$$

where ω_d is the rollup corner frequency/bandwidth. This stage has the following notable properties:

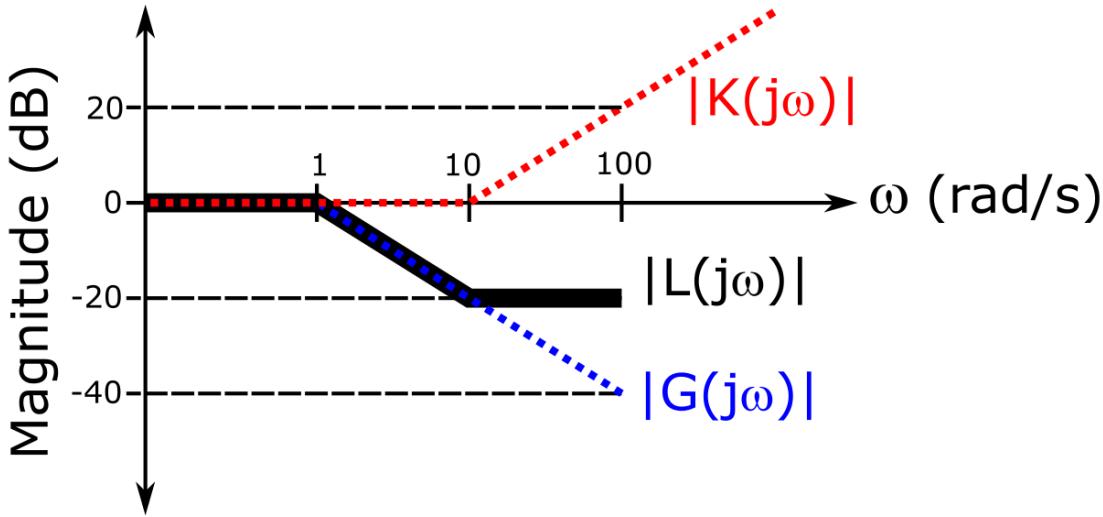
1. $K(s)$ has a zero at $s = -\omega_d$.
2. At low ω , $K(j\omega) \approx 1$ or 0 dB.
3. At high ω , $K(j\omega) \approx \frac{j\omega}{\omega_d}$, i.e. for every $10\omega \rightarrow 10K(j\omega)$ or a 20 dB/10ω slope.

As an example, consider the following

$$G(s) = \frac{1}{s + 1} \quad (2.124)$$

$$K(s) = 0.1s + 1 \quad (\omega_d = 10) \quad (2.125)$$

$$L(s) = 0.1 \frac{s + 10}{s + 1} \quad (2.126)$$



Low-Pass Filter Control Stage

This control stage is used to decrease the gain of $L(j\omega)$ for the high-frequency region $\omega > \omega_{l-p}$, i.e. to have noise filtering at high frequencies. This stage is “low-pass” as it passes frequencies below ω_{l-p} with approximately unity gain, i.e. “low-pass,” but significantly reduces the gain for higher frequencies. The types of low-pass filters used in practice can vary and include

- Butterworth filter
- Chebyshev filter
- Legendre-Papoulis filter
- Cauer filter
- Bessel filter

which balance the “unity gain” approximation and the attenuation or “rolloff” at the corner frequency. The order of these low-pass filter is typically a parameter as well where lower orders are typically preferred.

For SISO loop-shaping control design, one typically desires to keep a linear phase response, so low-pass Butterworth filters are often preferred. The n^{th} -order low-pass Butterworth filter control stage can be defined as

$$K_n(s) = \frac{1}{B_n \left(\frac{s}{\omega_{l-p}} \right)} \quad (2.127)$$

where ω_{l-p} is the filter corner frequency/bandwidth and B_n is the **normalized Butterworth polynomial** defined as

$$B_n(s) = \begin{cases} \prod_{k=0}^{\frac{n}{2}-1} \left(s^2 - 2 \cos \left(2\pi \frac{2k+n+1}{4n} \right) s + 1 \right) & n \text{ even} \\ (s+1) \prod_{k=0}^{\frac{n-1}{2}-1} \left(s^2 - 2 \cos \left(2\pi \frac{2k+n+1}{4n} \right) s + 1 \right) & n \text{ odd} \end{cases} \quad (2.128)$$

It can be shown that the magnitude of an n^{th} -order low-pass Butterworth filter is

$$\|K_n(j\omega)\| = \frac{1}{\sqrt{1 + \left(\frac{\omega}{\omega_{l-p}}\right)^{2n}}} \quad (2.129)$$

The transfer function for a first-order Butterworth filter would be

$$K(s) = \frac{\omega_{l-p}}{s + \omega_{l-p}} = \frac{1}{\frac{1}{\omega_{l-p}}s + 1} \quad (2.130)$$

where ω_{l-p} is the filter corner frequency/bandwidth. This stage has the following notable properties:

1. $K(s)$ has a pole at $s = -\omega_{l-p}$.
2. At low ω , $K(j\omega) \approx 1$ or 0 dB.
3. At high ω , $K(j\omega) \approx \frac{\omega_{l-p}}{j\omega}$, i.e. for every $10\omega \rightarrow K(j\omega)/10$ or a -20 dB/10 ω slope.

The transfer function for a second-order Butterworth filter would be

$$K(s) = \frac{\omega_{l-p}^2}{s^2 + \sqrt{2}\omega_{l-p}s + \omega_{l-p}^2} \quad (2.131)$$

where ω_{l-p} is the filter corner frequency/bandwidth and the damping ratio of the second-order system is $\sqrt{2}/2$. This stage has the following notable properties:

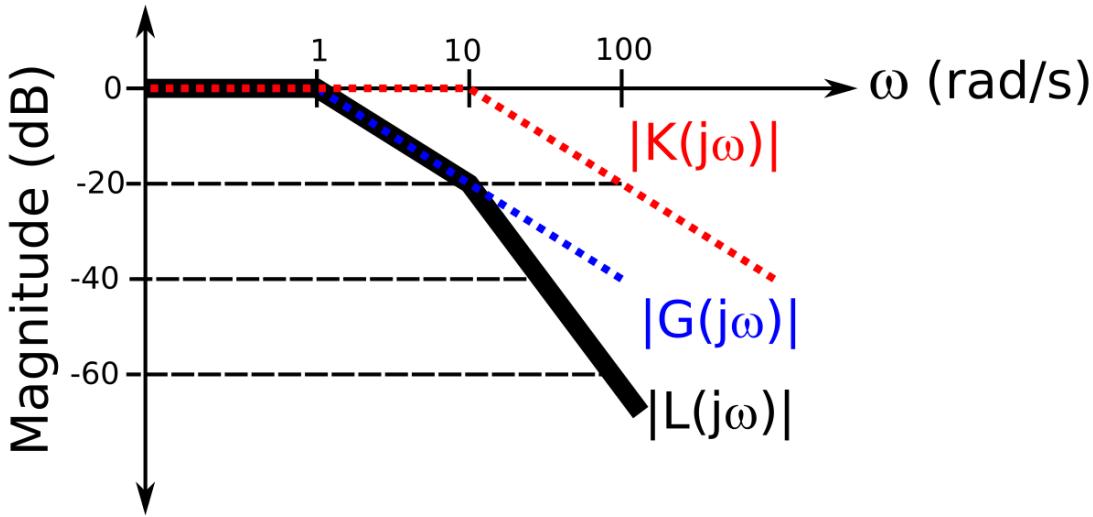
1. $K(s)$ has poles at $s = -\frac{\sqrt{2}}{2}\omega_{l-p} \pm j\frac{\sqrt{2}}{2}\omega_{l-p}$.
2. At low ω , $K(j\omega) \approx 1$ or 0 dB.
3. At high ω , $K(j\omega) \approx \frac{\omega_{l-p}}{-\omega^2}$, i.e. for every $10\omega \rightarrow K(j\omega)/100$ or a -40 dB/10 ω slope.

As an example, consider the following first-order low-pass Butterworth filter

$$G(s) = \frac{1}{s + 1} \quad (2.132)$$

$$K(s) = \frac{1}{B_n\left(\frac{s}{10}\right)} = \frac{10}{s + 10} \quad (2.133)$$

$$L(s) = \frac{10}{(s^2 + 11s + 10)} \quad (2.134)$$



High-Pass Filter Control Stage

Band-Pass Filter Control Stage

Low-Frequency Boost Control Stage

This control stage is used to alternatively increase the gain of $L(s)$ for the low-frequency region $\omega < \omega_i$, i.e. to have good tracking at low frequencies, without an infinite gain at $\omega = 0$ which may not be practical in hardware implementation. The transfer function for this control stage is defined as

$$K(s) = \frac{s + \omega_i}{s + \frac{\omega_i}{\beta}} \quad (2.135)$$

where ω_i is the corner frequency/bandwidth and $\beta > 1$ is the low frequency gain. This stage has the following notable properties:

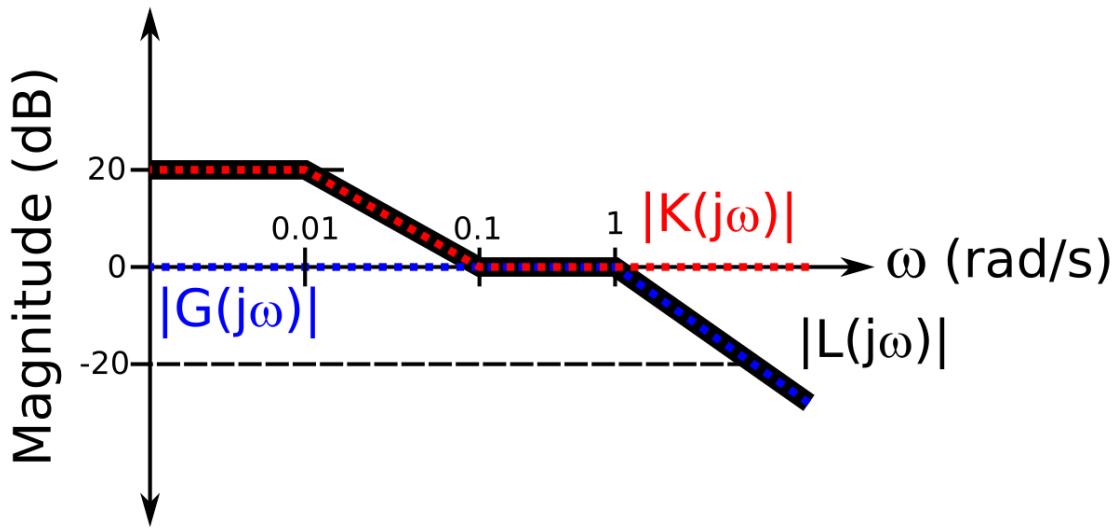
1. $K(s)$ has a pole at $s = -\frac{\omega_i}{\beta}$ and a zero at $-\omega_i$;
2. at low ω , $K(j\omega) \approx \beta$ or $20 \log_{10} \beta$ dB;
3. at high ω , $K(j\omega) \approx 1$ or 0 dB; and
4. $K(s)$ is similar to integral stage with ω_i , but levels off at the low frequency gain β , i.e. as $\beta \rightarrow \infty$, this stage approaches an integral stage.

As an example, consider the following

$$G(s) = \frac{1}{s + 1} \quad (2.136)$$

$$K(s) = \frac{s + 0.1}{s + 0.01} \quad (\omega_i = 0.1, \beta = 10) \quad (2.137)$$

$$L(s) = \frac{s + 0.1}{(s^2 + 1.01s + 0.01)} \quad (2.138)$$



Lead-Lag Control Stage

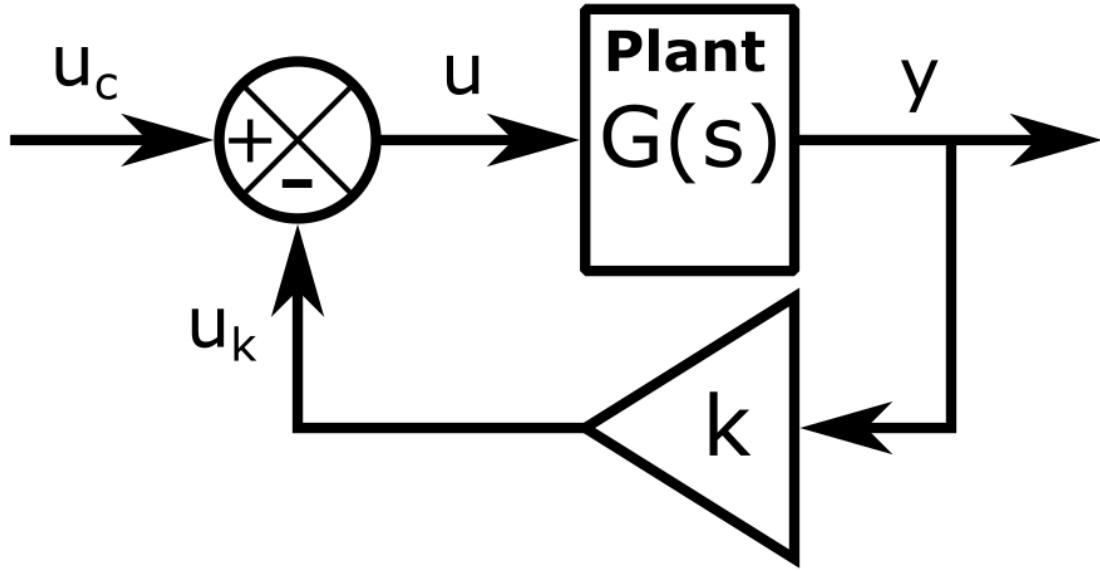
2.6 SAS and PID Control Design

Stability Augmentation System Design

The classical stability augmentation system design method is the root locus method defined as a **proportional SAS (P-SAS)** as

$$K_{P-SAS}(s) = \frac{u_k(s)}{y(s)} = k \quad (2.139)$$

where $k > 0$ is the negative feedback gain as shown in the following block diagram



Thus, a P-SAS produces an overall system transfer function

$$y(s) = \left[\frac{G(s)}{1 + kG(s)} \right] u_c(s) \quad (2.140)$$

with closed-loop system poles given by

$$d_G(s) + kn_G(s) = 0 \quad (2.141)$$

where $n_G(s)$ is the numerator polynomial of $G(s)$ and $d_G(s)$ is the denominator polynomial of the minimum-phase $G(s)$. Thus, by sweeping through values of k from 0 to ∞ , one can alter the system poles from $d_G(s)$ to $n_G(s)$ and/or $\pm\infty$, depending on the number of zeros relative to the number of poles. This behavior can easily be plotted on the **root locus** plot of the system poles and zeros in the complex plane as k varies from 0 to ∞ , e.g. if one has a second-order plant given by

$$G(s) = \frac{s+2}{s^2 + 2s + 2} \quad (2.142)$$

which has poles at $s = -1 \pm j$ and a zero at $s = -2$.

Then, for a P-SAS system, the closed-loop characteristic equation is

$$s^2 + 2s + 2 + k(s+2) = 0 \quad (2.143)$$

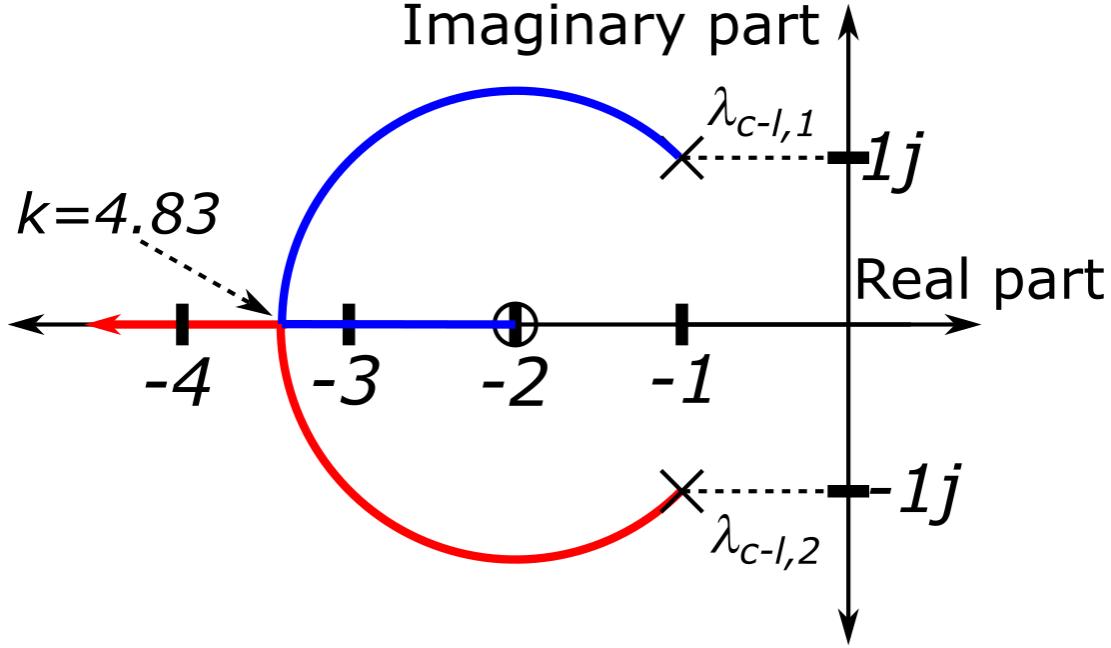
or

$$s^2 + (k+2)s + (2k+2) = 0 \quad (2.144)$$

for which the two closed-loop poles, $\lambda_{c-l,1}$ and $\lambda_{c-l,2}$, are given by

$$\lambda_{c-l,1}, \lambda_{c-l,2} = \frac{-(k+2) \pm \sqrt{(k+2)^2 - 4(2k+2)}}{2} \quad (2.145)$$

Then, one has a root locus given by



where one can place $\lambda_{c-l,1}$ and $\lambda_{c-l,2}$ anywhere along the blue or red lines, respectively, depending on the choice of k .

Notably, the second-order closed-loop system becomes critically damped, $\zeta = 1$, at

$$k + 2 = 2\sqrt{2k + 2} \quad (2.146)$$

$$k^2 + 4k + 4 = 8k + 8 \quad (2.147)$$

$$k^2 - 4k - 4 = 0 \quad (2.148)$$

$$k = \frac{4 \pm \sqrt{(-4)^2 - 4(-4)}}{2} = 2 \pm 2\sqrt{2} \quad (2.149)$$

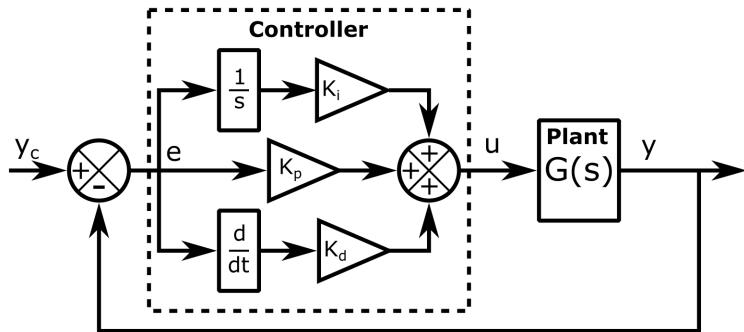
$$k = -0.83 \text{ or } 4.83 \quad (2.150)$$

Proportional-Integral-Derivative Control Design

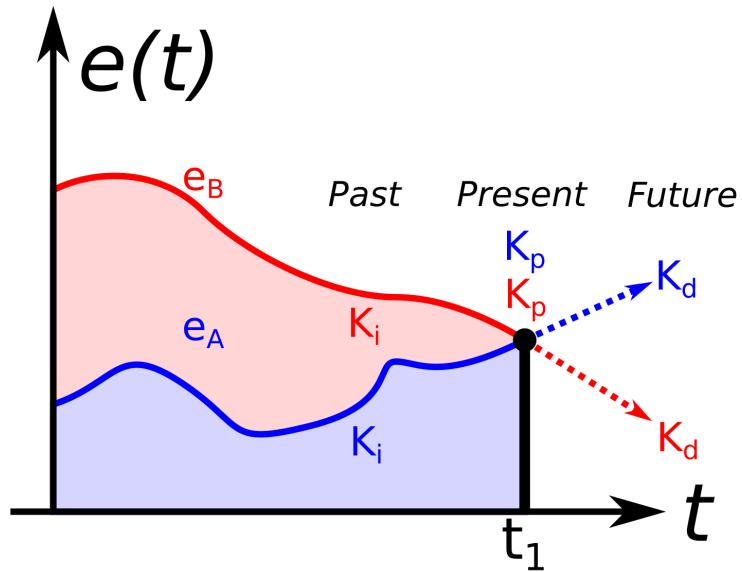
The most common classical control design method is the **proportional-integral-derivative (PID)** control law defined in **parallel form** as

$$K_{PID}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d s \quad (2.151)$$

where K_p is the **proportional gain**, K_i is the **integral gain**, and K_d is the **derivative gain**.



It should be noted that if $K_d = 0$, one has a proportional-integral (PI) controller, if $K_i = 0$, one has a proportional-derivative (PD) controller and if $K_d = K_i = 0$, one has a simple proportional (P) controller. Here the control designer must choose the set of gains, (K_p, K_i, K_d) , to satisfy the design requirements. An intuitive understanding of PID control is to look at the following diagram of the tracking error history for two different systems as shown below.



Here the proportional term would notably have the same correction effect here at the present time, t_1 , while the integral term has a correction effect based on the accumulated *past* error and continues to grow until the controller reaches u_{ss} for $e_{ss} = 0$. Lastly, the derivative term has a correction effect based on the *future* trend of the tracking error, i.e. its rate of change, and thus can improve the speed of response and reduce the damping.

A common alternative form for PID control design is the **standard form**

$$K_{PID}(s) = \frac{u(s)}{e(s)} = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d s \right) \quad (2.152)$$

where $\tau_i = \frac{K_p}{K_i}$ is the **integral time** and $\tau_d = \frac{K_d}{K_p}$ is the **derivative time**. However, for PID control design via loop-shaping, one typically considers the **series form**, also known as the **interacting form**, of the PID controller, i.e.

$$K_{PID}(s) = k \left(\frac{s + \omega_i}{s} \right) \left(\frac{1}{\omega_d} s + 1 \right) \quad (2.153)$$

where k is the overall gain, ω_i is the integral frequency, and ω_d is the derivative frequency. This form demonstrates PID control as a particular loop-shaping controller employing three control stages: a proportional gain of k , an integral boost at ω_i , and a derivative stage at ω_d . These three stages can be used to shape $L(s)$ by choosing k for the loop bandwidth, ω_i for the low frequency performance requirements, and ω_d for the stability requirements. It should be noted that k is typically limited by the control effort requirements.

Furthermore, the PID series form design parameters can be related to the PID standard form design parameters by

$$K_p = k \left(1 + \frac{\omega_i}{\omega_d} \right) \quad (2.154)$$

$$\tau_i = \frac{1}{\omega_i} + \frac{1}{\omega_d} \quad (2.155)$$

and

$$\tau_d = \frac{1}{\omega_i + \omega_d} \quad (2.156)$$

Noting the derivative boost stage, one can see that the derivative term increases the asymptotic slope of the open-loop transfer function, $L(s)$, which will amplify any high frequency sensor noise in y_m in the classical feedback control system. To overcome this unfavorable outcome, often the derivative term in PID control is alternatively implemented using a **filtered derivative**. This filtered PID (PIDF) controller can be defined in **PIDF parallel form** as

$$K_{PIDF}(s) = \frac{u(s)}{e(s)} = K_p + \frac{K_i}{s} + K_d \frac{s}{\frac{1}{\omega_\infty} s + 1} \quad (2.157)$$

or, alternatively, in **PIDF standard form**

$$K_{PIDF}(s) = K_p \left(1 + \frac{1}{\tau_i s} + \tau_d \frac{s}{\frac{1}{\omega_\infty} s + 1} \right) \quad (2.158)$$

where ω_∞ is the bandwidth of the “filtered” derivative, i.e. as $\omega_\infty \rightarrow \infty$, the transfer function $\frac{s}{\frac{1}{\omega_\infty} s + 1}$ approaches a pure differentiator. Thus, one can rewrite this type of PID controller in **PIDF series form** as

$$K_{PIDF}(s) = k \left(\frac{s + \omega_i}{s} \right) \left(\frac{\beta^2}{\omega_\infty} s + 1 \right) \left(\frac{\omega_\infty}{s + \omega_\infty} \right) \quad (2.159)$$

where the series form design parameters can be related to the standard form design parameters by

$$K_p = k \frac{(\beta^2 - 1)\omega_i}{\omega_\infty} + 1 \quad (2.160)$$

$$\tau_i = \frac{\beta^2 - 1}{\omega_\infty} + \frac{1}{\omega_i} \quad (2.161)$$

and

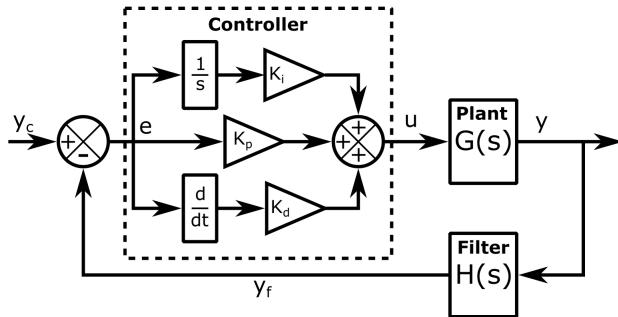
$$\tau_d = \frac{\beta^2}{((\beta^2 - 1)\omega_i + \omega_\infty)} - \frac{1}{\omega_\infty} \quad (2.162)$$

Equivalently, by the substitution $\omega_d = \omega_\infty/\beta^2$, one has

$$K_{PIDF}(s) = k \left(\frac{s + \omega_i}{s} \right) \left(\frac{1}{\omega_d} s + 1 \right) \left(\frac{\omega_\infty}{s + \omega_\infty} \right) \quad (2.163)$$

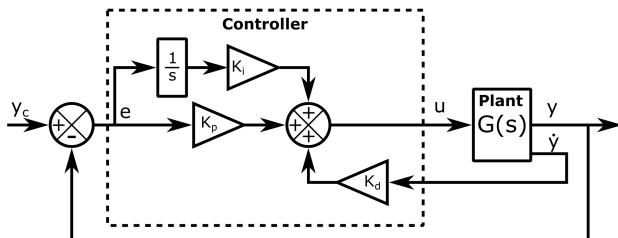
This form demonstrates filtered PID control has a particular loop-shaping controller employing four control stages: a proportional with k , an integral at ω_i , a derivative stage at ω_d , and a first-order filter at ω_∞ . Thus, these stages allow one to use the loop-shaping design procedure to iterate on the values of these control stages to affect the low-frequency, crossover, and high-frequency regions of the $L(j\omega)$, while also implementing them as classical parallel PID control gains.

It should be noted that while PID controllers use zero or one low-pass filter control stages, one may be required to use additional filter stages in addition to the PID controller, though often these are implemented as part of the filter subsystem for y_f from y_m as shown in the following block diagram,



Note that one can also limit the update rate, i.e. frequency, of y_c , due to control signal sampling limitations or previous filtering.

As another alternative, one often can measure the output derivative directly instead of computing the derivative of the tracking error signal, so often the derivative term in PID control is alternatively implemented using **rate feedback control** where the derivative of the output, i.e. “rate,” is fed back. Thus, **PI control with rate feedback** would implement a system



This may be understood as a SIMO feedback control system as two separate outputs are fed back to the controller, the error and the output rate. It may also be considered as a **stability augmentation system (SAS)** using the rate and a second closed-loop PI controller using the output error.

Ziegler-Nichols PID Tuning Method

The **Ziegler–Nichols (Z-N) PID tuning method** was developed by John Ziegler and Nathaniel Nichols as a simple heuristic method of tuning the gains of a PID-esque controllers which can be applied for any SISO system to maximize the disturbance rejection capabilities of PID. The Z-N PID tuning method works well when you have an analog controller for a stable LTI system whose response is dominated by a single stable mode. Though actual plants are unlikely to have only a first-order pole, this approximation is reasonable to describe the frequency response rolloff in many instances. Higher-order poles will introduce an extra phase shift, especially if the dominate mode is a second-order underdamped mode. Thus, the loop-shaping approach is typically the most flexible for LTI system design. These have also been adjusted by different control system designers into a few different rule sets. These methods are performed by first setting the K_i and K_d gains to zero, then increasing K_p from zero to the *ultimate gain*, $K_p = K_u$, which is defined as the gain for which the output of the feedback control system has stable and consistent oscillations of period, T_u , or as the inverse of the gain margin for an LTI plant.

Then, one can set the P, PI, PID gains according to the following table for either the classic or standard forms.

| Controller Type | K_p | τ_i | τ_d | K_i | K_d |
|--------------------------|----------------|-----------------|----------------|---------------|---------------------|
| P | $0.5K_u$ | - | - | - | - |
| PI | $0.45K_u$ | $0.8\bar{3}T_u$ | - | $0.54K_u/T_u$ | - |
| PD | $0.8K_u$ | - | $0.125T_u$ | - | $0.0\bar{6}K_u/T_u$ |
| Classic PID | $0.6K_u$ | $0.5T_u$ | $0.125T_u$ | $1.2K_u/T_u$ | $0.0\bar{6}K_u/T_u$ |
| Pessen Integral Rule PID | $0.7K_u$ | $0.4T_u$ | $0.15T_u$ | $1.75K_u/T_u$ | $0.0\bar{6}K_u/T_u$ |
| Some Overshoot PID | $0.\bar{3}K_u$ | $0.5T_u$ | $0.\bar{3}T_u$ | $0.6K_u/T_u$ | $0.1K_u/T_u$ |
| No Overshoot PID | $0.2K_u$ | $0.5T_u$ | $0.\bar{3}T_u$ | $0.4K_u/T_u$ | $0.0\bar{6}K_u/T_u$ |

References

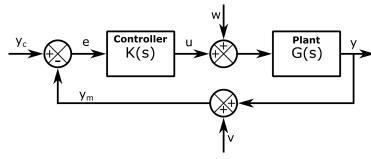
For more information, please refer to the following

- McCormack, A. S., and Godfrey, K. R., “Rule-Based Autotuning Based on Frequency Domain Identification,” *IEEE Transactions on Control Systems Technology*, vol, 6 no. 1, January 1998
- Nelson, R. C., “7.7 Control System Design” and “7.8 PID Controller,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 262-274
- Schmidt, D. K., “12.1 Feedback Control-Law Synthesis Via Loop Shaping - A Just-In-Time Tutorial*,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 548-562

- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “3.9 Feedback Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 213-240

2.7 Classical Feedback Control System Performance Constraints

LTI feedback control systems must satisfy robustness requirements for stability *as well as* performance requirements. Thus, it is important to be aware of fundamental limits on the performance of the control system. To this end, recall the block diagram for a SISO LTI feedback control system.



Also, recall the error/sensitivity transfer function can be defined as

$$e(s) = S(s)y_c(s) \quad (2.164)$$

and the closed-loop/complementary sensitivity transfer function is defined as

$$y(s) = T(s)y_c(s) \quad (2.165)$$

where

$$S(s) + T(s) = 1 \quad \forall s \quad (2.166)$$

Performance Limits from Sensitivity Tradeoff

One can deduce performance limits on the closed-loop response in the frequency domain by analyzing the Bode sensitivity integral theorem whose proof ultimately relies on Cauchy’s integral theorem which is a result from complex analysis. This theorem states that if $1 + L(s)$ has no zeros in the RHP and $L(s)$ has relative degree ≥ 2 , i.e. the denominator is at least two orders higher than the numerator, then

$$\int_0^\infty \log_{10} |S(j\omega)| d\omega = \pi \sum_{i=1}^{N_p} \operatorname{Re}\{p_i\} \quad (2.167)$$

where p_i for $i = 1, \dots, N_p$ are the N_p open RHP poles of $L(s)$. Furthermore, if $L(s)$ has no poles in the RHP, then one has

$$\int_0^\infty \log_{10} |S(j\omega)| d\omega = 0 \quad (2.168)$$

which illustrates the **sensitivity tradeoff** that any decrease in $S(j\omega)$ must be balanced by an increase in $S(j\omega)$, i.e. the area under the curve of the Bode plot of $S(j\omega)$ must be conserved.

Note that if there exist poles in the RHP, the area of sensitivity increase will exceed that of the sensitivity decrease. Furthermore, though this result does not depend on RHP zeros of $G(s)$, there are related results based on Poisson integrals which demonstrate that RHP zeros will further degrade this sensitivity tradeoff. In addition, note that the restriction for $L(s) = G(s)K(s)$ to have relative degree ≥ 2 is essentially always satisfied in practice. $G(s)$ will be strictly proper, i.e. its denominator will have higher order than its numerator, as all real plants eventually rolloff at high frequencies. Similarly, $K(s)$ will also be strictly proper since it will be implemented on hardware whose dynamics rolloff at high frequencies.

An important part of the control design process is to set $|S(j\omega)| \ll 1$ for low frequencies. This can be more formally stated as some limit on $\log_{10} |S(j\omega)| \leq \alpha \ll 0$ for all $\omega \leq \omega_L$ for some ω_L . However, to push down $\log_{10} |S(j\omega)|$, by the Bode sensitivity integral, one has to increase the area above 0 dB. Thus, one may think the best approach is to shape the entire area of $|S(j\omega)|$ to have some small amount over all frequencies, $\omega > \omega_L$. However, in practice, this is not possible as one does not have an accurate model of $G(j\omega)$ at high frequencies and must be robust to this consideration. More formally stated, the control design process typically has as some upper limit on the **available bandwidth**, ω_U , where $|L(j\omega)|$ rolls off rapidly for $\omega \geq \omega_U$, i.e.

$$|L(j\omega)| \leq \epsilon \frac{\omega_U^2}{\omega} \quad \text{for all } \omega \geq \omega_U \quad (2.169)$$

for some $\epsilon < 0.5$. This practical consideration of ω_U allows one to split the Bode sensitivity integral for no RHP poles into three pieces

$$\int_0^{\omega_L} \log_{10} |S(j\omega)| d\omega + \int_{\omega_L}^{\omega_U} \log_{10} |S(j\omega)| d\omega + \int_{\omega_U}^{\infty} \log_{10} |S(j\omega)| d\omega = 0 \quad (2.170)$$

or

$$-\left(\int_0^{\omega_L} \log_{10} |S(j\omega)| d\omega + \int_{\omega_U}^{\infty} \log_{10} |S(j\omega)| d\omega \right) = \int_{\omega_L}^{\omega_U} \log_{10} |S(j\omega)| d\omega \quad (2.171)$$

Then, assuming $L(s)$ is stable and has relative degree ≥ 2 where each can be bounded by

$$-\int_0^{\omega_L} \log_{10} |S(j\omega)| d\omega \geq \omega_L \log_{10} \frac{1}{\alpha} \quad (2.172)$$

$$-\int_{\omega_U}^{\infty} \log_{10} |S(j\omega)| \geq -\frac{3\epsilon}{2} \omega_c \quad (2.173)$$

$$\int_{\omega_L}^{\omega_U} \log_{10} |S(j\omega)| d\omega \leq (\omega_U - \omega_L) \left(\max_{\omega \in [\omega_L, \omega_U]} \log_{10} |S(j\omega)| \right) \quad (2.174)$$

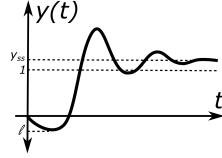
leads to the following

$$\max_{\omega \in [\omega_L, \omega_U]} \log_{10} |S(j\omega)| \geq \frac{\omega_L}{\omega_U - \omega_L} \log_{10} \frac{1}{\alpha} - \left(\frac{3\epsilon}{2} \right) \frac{\omega_c}{\omega_U - \omega_L} \quad (2.175)$$

which is also known as the **waterbed effect** which derives from the tradeoff between “pushing down” on $S(s)$ causing it to “rise up” somewhere between $\omega_L \leq \omega \leq \omega_U$ with the peak value, $\max |S(s)| \rightarrow \infty$ as $\alpha \rightarrow 0$. Note that this formulation did not include RHP poles and zeros which will make the waterbed effect even worse.

Performance Limit Due to Real RHP Zero

Assume that $G(s)$ has a real RHP zero at $s = z$. This will cause the feedback control system response, $y(t)$, to initially undershoot by some amount, ℓ , which can be visualized as



and defined as

$$\ell = \min_{t \geq 0} y(t) \quad (2.176)$$

If there is a zero, z , in the closed RHP such that $G(z) = 0$, then one has $G(z)K(z) = 0$ as there can be no pole/zero cancellations, thus the closed-loop transfer function is then

$$T(z) = \frac{G(z)K(z)}{1 + G(z)K(z)} = 0 \quad (2.177)$$

Thus, the unit step response has a transfer function at $s = z$ that satisfies

$$y(z) = T(z)y_c(z) = 0 \quad (2.178)$$

From the definition of the Laplace transform, one has

$$\int_0^\infty e^{-zt} y(t) dt = 0 \quad (2.179)$$

which can be split into two pieces

$$\int_0^{t_{s,w}} e^{-zt} y(t) dt + \int_{t_{s,w}}^\infty e^{-zt} y(t) dt = 0 \quad (2.180)$$

where the settling time, $t_{s,w}$, to some percentage w for the unit step response is defined as

$$t_{s,w} = \min t \quad \text{such that} \quad |y(t) - y_{ss}| \leq 0.05y_{ss} \quad t \geq t_{s,w} \quad (2.181)$$

where y_{ss} is the steady-state output, i.e. $y(t)$ as $t \rightarrow \infty$.

Thus,

$$\int_{t_{s,w}}^\infty e^{-zt} y(t) dt = - \int_0^{t_{s,w}} e^{-zt} y(t) dt \quad (2.182)$$

By the definition of the minimum of $y(t)$, one has

$$-\int_0^{t_{s,w}} e^{-zt} y(t) dt \geq -\ell \int_0^{t_{s,w}} e^{-zt} dt \quad (2.183)$$

or

$$\ell \leq \frac{-\int_{t_{s,w}}^{\infty} e^{-zt} y(t) dt}{\int_0^{t_{s,w}} e^{-zt} dt} \quad (2.184)$$

For $t > t_{s,w}$, one has $y(t) \geq (1-w)y_{ss}$ or $-y(t) \leq -(1-w)y_{ss}$ and one can bound the undershoot by

$$\ell \leq \frac{-(1-w) \int_{t_{s,w}}^{\infty} e^{-zt} dt}{\int_0^{t_{s,w}} e^{-zt} dt} \quad (2.185)$$

which by integration for a real-valued z , one has

$$\ell \leq \frac{-(1-w) [\frac{1}{z} (e^{-zt_{s,w}})]}{[\frac{1}{z} (1 - e^{-zt_{s,w}})]} \quad (2.186)$$

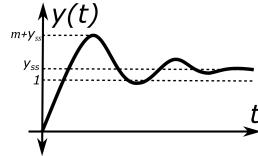
or

$$\ell \leq \frac{-(1-w)}{e^{zt_{s,w}} - 1} \quad (2.187)$$

Thus, for a fixed zero, z , for the plant, $G(s)$, as one changes $K(s)$ to decrease the settling time, one must get a larger amount of undershoot, i.e. as $t_{s,w} \rightarrow 0$, $\ell \rightarrow -\infty$, or conversely, to decrease the undershoot, one must increase the settling time, $t_{s,w}$. In general, a RHP zero, even if complex, places an upper bound on the bandwidth of the classical feedback control system.

Performance Limit Due to Real RHP Pole

Assume that $G(s)$ has a real RHP pole at $s = p$. This will cause the feedback control system response, $y(t)$, to have some overshoot by some amount, m , which can be visualized as



and defined as

$$m = \max_{t \geq 0} y(t) - y_{ss} \quad (2.188)$$

where $y_{ss} = 1$ for simplicity.

If there is a pole, p , in the closed RHP such that $G(p) = \infty$, then $G(p)K(p) = \infty$ as there can be no pole/zero cancellations, thus

$$S(p) = \frac{1}{1 + G(p)K(p)} = 0 \quad (2.189)$$

Thus, the tracking error has a transfer function at $s = p$ that satisfies

$$e(p) = S(p)y_c(p) = 0 \quad (2.190)$$

From the definition of the Laplace transform, one has

$$\int_0^\infty e^{-pt} e(t) dt = 0 \quad (2.191)$$

which can be split into two pieces

$$\int_0^{t_{r,v}} e^{-pt} e(t) dt + \int_{t_{r,v}}^\infty e^{-pt} e(t) dt = 0 \quad (2.192)$$

where the rise time, $t_{r,v}$, for the unit step response is defined as

$$t_{r,v} = \min t \text{ such that } y(t) \leq vy_{ss} \text{ for all } t \leq t_{r,v} \quad (2.193)$$

Thus, as $e(t) = 1 - y(t)$ for the unit step, one has

$$\int_{t_{r,v}}^\infty e^{-pt} e(t) dt = - \int_0^{t_{r,v}} e^{-pt} e(t) dt \quad (2.194)$$

By the definition of the overshoot for $e(t)$, one has

$$\int_{t_{r,v}}^\infty e^{-pt} e(t) dt \geq (1 - m - y_{ss}) \int_{t_{r,v}}^\infty e^{-pt} dt = -m \left[\frac{1}{p} e^{-pt_{r,v}} \right] \quad (2.195)$$

and by the definition of the rise time for $e(t)$

$$-\int_0^{t_{r,v}} e^{-pt} e(t) dt \leq -(1 - v) \int_0^{t_{r,v}} e^{-pt} dt = -(1 - v) \left[\frac{1}{p} (1 - e^{-pt_{r,v}}) \right] \quad (2.196)$$

Combining these two bounds, one has

$$-m \left[\frac{1}{p} e^{-pt_{r,v}} \right] \leq -(1 - v) \left[\frac{1}{p} (1 - e^{-pt_{r,v}}) \right] \quad (2.197)$$

or

$$m \geq (1 - v)(e^{-pt_{r,v}} - 1) \quad (2.198)$$

Thus, for a fixed pole, p , for the plant, $G(s)$, as one changes $K(s)$ to increase the rise time, one must get a larger amount of overshoot, i.e. as $t_{r,v} \rightarrow \infty$, $m \rightarrow \infty$, or conversely, to decrease the overshoot, one must decrease the rise time, $t_{r,v}$. In general, a RHP pole places a lower bound on the bandwidth of the classical feedback control system. However, it should be noted that this limit assumes that the controller only receives the error signal $e(t)$. However, this can be avoided for MIMO inputs to the controller.

Performance Limits Due to Real RHP Zero and Pole

Assume that $G(s)$ has both a real RHP zero at $s = z$ and a real RHP pole at $s = p$. This will cause the feedback control system response, $y(t)$, to have some undershoot by some amount ℓ , and some overshoot by some amount, m . In this case, the unit step response has a transfer function at $s = z$ that satisfies

$$y(z) = T(z)y_c(z) = 0 \quad (2.199)$$

and the tracking error has a transfer function at $s = p$ that satisfies

$$e(p) = S(p)y_c(p) = 0 \quad (2.200)$$

From the definition of the Laplace transform, one has

$$\int_0^\infty e^{-zt} y(t) dt = 0 \quad (2.201)$$

and

$$\int_0^\infty e^{-pt} e(t) dt = 0 \quad (2.202)$$

Furthermore, as $e(t) = 1 - y(t)$, one has

$$\int_0^\infty e^{-zt} (1 - e(t)) dt = 0 \quad (2.203)$$

and

$$\int_0^\infty e^{-pt} (1 - y(t)) dt = 0 \quad (2.204)$$

Next, by rearranging, one has

$$\int_0^\infty e^{-zt} dt = \int_0^\infty e^{-zt} e(t) dt \quad (2.205)$$

and

$$\int_0^\infty e^{-pt} dt = \int_0^\infty e^{-pt} y(t) dt \quad (2.206)$$

and computing the integrals on the left sides, one has

$$\frac{1}{z} = \int_0^\infty e^{-zt} e(t) dt \quad (2.207)$$

and

$$\frac{1}{p} = \int_0^\infty e^{-pt} y(t) dt \quad (2.208)$$

Next, subtracting these from the original $e(p)$ and $y(z)$ integrals, one has

$$\int_0^\infty e^{-zt} y(t) dt - \int_0^\infty e^{-pt} y(t) dt = 0 - \frac{1}{p} \quad (2.209)$$

and

$$\int_0^\infty e^{-pt} e(t) dt - \int_0^\infty e^{-zt} e(t) dt = 0 - \frac{1}{z} \quad (2.210)$$

or

$$\int_0^\infty (e^{-zt} - e^{-pt}) y(t) dt = -\frac{1}{p} \quad (2.211)$$

and

$$\int_0^\infty (e^{-pt} - e^{-zt}) e(t) dt = -\frac{1}{z} \quad (2.212)$$

By the definition of the minimum of $y(t)$, one has

$$\ell \int_0^\infty e^{-zt} - e^{-pt} dt \leq \int_0^\infty (e^{-zt} - e^{-pt}) y(t) dt = -\frac{1}{p} \quad (2.213)$$

and the definition of the overshoot for $e(t)$, one has

$$-m \int_0^\infty e^{-zt} - e^{-pt} dt \leq \int_0^\infty (e^{-zt} - e^{-pt}) e(t) dt = -\frac{1}{z} \quad (2.214)$$

Calculating the integrals, one has

$$\ell \frac{p-z}{zp} \leq -\frac{1}{p} \quad (2.215)$$

and

$$-m \frac{z-p}{zp} \leq -\frac{1}{z} \quad (2.216)$$

Thus, if $p > z$, then dividing by $\frac{p-z}{zp}$, one has

$$\ell \leq -\frac{z}{p-z} \quad (2.217)$$

and if $z > p$, then dividing by $-\frac{z-p}{zp}$, one has

$$m \geq \frac{p}{z-p} \quad (2.218)$$

In either case, it should be noted that if p is near to z , then the denominator is close to zero and the bound has large magnitude. This implies that plants with “close” RHP poles/zeros are difficult to control.

This analysis can be extended to the sensitivity transfer function. First, one must know the **maximum modulus theorem** which states for a stable transfer function, $G(s)$, the peak magnitude over the imaginary axis is equal to the peak magnitude over the entire RHP, i.e.

$$\max_{\omega \in \mathbb{R}} \|G(j\omega)\| = \max_{s \in \mathbb{C}, \operatorname{Re}(s) \geq 0} \|G(s)\| \quad (2.219)$$

Suppose the classical feedback system is stable and $G(s)$ has a real RHP zero at $s = z$ and a real RHP pole at $s = p$, then one has $S(p) = 0$ and $S(z) = 1$. Furthermore, consider factoring $S(s)$ as

$$S(s) = \frac{s-p}{s+z} \tilde{S}(s) \quad (2.220)$$

where $\tilde{S}(s)$ is stable and one can write

$$\max_{\omega} |S(j\omega)| = \max_{\omega} |j\omega - p j\omega + p \tilde{S}(j\omega)| = \max_{\omega} |\tilde{S}(j\omega)| \quad (2.221)$$

as

$$|j\omega - p j\omega + p| = 1 \quad \forall \omega \quad (2.222)$$

Then, by the maximum modulus theorem

$$\max_{\omega} |\tilde{S}(j\omega)| \geq |\tilde{S}(z)| \quad (2.223)$$

Finally, note that

$$1 = S(z) = \frac{z - p}{z + p} \tilde{S}(z) \quad (2.224)$$

or

$$\tilde{S}(z) = \frac{z + p}{z - p} \quad (2.225)$$

Thus, one can state

$$\max_{\omega} |S(j\omega)| \geq \left| \frac{z+p}{z-p} \right| \quad (2.226)$$

Then, recall that $\|S(j\omega)\|$ is inversely proportional to the distance between $L(s)$ and the critical $s = -1$ point for *any* $K(s)$, i.e. the disk margin. Recalling

$$d_{min} = \frac{1}{\max_{\omega} |S(j\omega)|} \quad (2.227)$$

one can also state

$$d_{min} \leq \left| \frac{z-p}{z+p} \right| \quad (2.228)$$

Thus, this result also demonstrates that plants with “close” RHP poles/zeros are difficult to achieve good robustness.

References

For more information, please refer to the following

- Skogestad, S., and Postlethwaite, I., “5 Limitations on Performance in SISO Systems,” *Multivariable Feedback Control: Analysis and Design*, 1st ed., Vol. 1, John Wiley & Sons, Chichester, England, 1996, pp. 159-212

Continuous-Time MIMO Linear Feedback Control Theory

3.1 MIMO LTI System Analysis

Initial Condition Response for MIMO LTI Systems

For an initial condition at $\vec{x}(0)$, the initial condition response, i.e. the zero-input or free response, of a continuous-time state-space system representation can be reduced to the following

$$\dot{\vec{x}}(t) = A \vec{x}(t) , \quad (3.1)$$

By using the JCF, i.e.

$$\vec{x}(t) = V \vec{z}(t) \quad (3.2)$$

where V is the matrix of n eigenvectors of A , then, the state-space representation can be transformed to using a new state, $\vec{z}(t)$ as

$$\dot{\vec{z}}(t) = \Lambda \vec{z}(t) \quad (3.3)$$

where Λ is in Jordan form, i.e. diagonal or nearly diagonal.

Assuming A is diagonalizable, then Λ is diagonal and the state-space is

$$\begin{bmatrix} \dot{z}_1(t) \\ \vdots \\ \dot{z}_n(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \lambda_n \end{bmatrix} \begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} \quad (3.4)$$

$$\begin{bmatrix} \dot{z}_1(t) \\ \vdots \\ \dot{z}_n(t) \end{bmatrix} = \begin{bmatrix} \lambda_1 z_1(t) \\ \vdots \\ \lambda_n z_n(t) \end{bmatrix} \quad (3.5)$$

Then, since each component is independent, the free response is given by the homogeneous solution to a first order ODE, i.e.

$$\begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} = \begin{bmatrix} e^{\lambda_1 t} z_1(0) \\ \vdots \\ e^{\lambda_n t} z_n(0) \end{bmatrix} \quad (3.6)$$

$$\begin{bmatrix} z_1(t) \\ \vdots \\ z_n(t) \end{bmatrix} = \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} \begin{bmatrix} z_1(0) \\ \vdots \\ z_n(0) \end{bmatrix} \quad (3.7)$$

$$\vec{z}(t) = \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} \vec{z}(0) \quad (3.8)$$

and using the inverse transformation, $\vec{z}(t) = V^{-1} \vec{x}(t)$, one can solve for the free response solution in the original state as

$$V^{-1} \vec{x}(t) = \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} V^{-1} \vec{x}(0) \quad (3.9)$$

$$\vec{x}(t) = V \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} V^{-1} \vec{x}(0) \quad (3.10)$$

$$\vec{x}(t) = [\vec{v}_1 \ \cdots \ \vec{v}_n] \begin{bmatrix} e^{\lambda_1 t} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & e^{\lambda_n t} \end{bmatrix} V^{-1} \vec{x}(0) \quad (3.11)$$

$$\vec{x}(t) = [e^{\lambda_1 t} \vec{v}_1 \ \cdots \ e^{\lambda_n t} \vec{v}_n] V^{-1} \vec{x}(0) \quad (3.12)$$

and representing

$$V^{-1} \vec{x}(0) = \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (3.13)$$

where c_1, \dots, c_n are scalar constants, one has

$$\vec{x}(t) = [e^{\lambda_1 t} \vec{v}_1 \ \cdots \ e^{\lambda_n t} \vec{v}_n] \begin{bmatrix} c_1 \\ \vdots \\ c_n \end{bmatrix} \quad (3.14)$$

$$\vec{x}(t) = c_1 e^{\lambda_1 t} \vec{v}_1 + \dots + c_n e^{\lambda_n t} \vec{v}_n \quad (3.15)$$

which is a similar form to the SISO LTI free response except here the eigenvectors have an additional affect. By analyzing the relative strength of each element in an eigenvector, it is simpler to observe which eigenvalues

affect which states more than others. If some eigenvalues/eigenvectors are complex conjugate pairs, then this expression can be rewritten using sin and cos functions to get only the real part of the solution for $x(t)$. This type of analysis is often called **modal analysis** where each real eigenvalue or complex conjugate eigenvalue pair denotes one **mode** of the solution.

It is also important to note that if A is not diagonalizable, the general solution can be found using **Jordan Chains** which are related to the generalized eigenvectors for repeated eigenvalues. This will include t, t^2, \dots, t^{n-1} terms and is quite similar to the SISO LTI free response for repeated roots of the characteristic equation. This result is left to other linear algebra resources.

In particular, continuous-time LTI systems are globally stable if and only if the eigenvalues of A are in the closed left half of the complex plane (LHP), i.e. the real part of the eigenvalues ≤ 0 . As a partial proof, consider the free response

$$\vec{x}(t) = c_1 e^{\lambda_1 t} \vec{v}_1 + \dots + c_n e^{\lambda_n t} \vec{v}_n \quad (3.16)$$

as $t \rightarrow \infty$, each i^{th} component with:

- $\text{Real}(\lambda_i) < 0$ will $\rightarrow 0$
- $\text{Real}(\lambda_i) = 0$ will remain constant if λ_i is purely real
- $\text{Real}(\lambda_i) = 0$ will oscillate with a constant magnitude if λ_i is purely imaginary

all of which are bounded (i.e. there exists some ϵ). A full proof must include non-diagonalizable A and can be done in a similar fashion.

Each of these stabilities can be expanded to include two different types. Continuous-time MIMO LTI systems are **marginally stable** if $\text{real}(\lambda_i) \leq 0 \forall i$ with $\text{real}(\lambda_j) = 0$ for at least one index j and are **asymptotically stable** if all $\text{real}(\lambda_i) < 0 \forall i$.

Frequency Response of MIMO LTI Systems

Recall the frequency response of a SISO LTI system is the Fourier transform of the system, i.e. $G(j\omega)$, which, by the Fourier inversion theorem, provides the system response to *all* harmonic sinusoids within any “well-behaved” input signal and from which one can exactly replicate the output signal. Thus, the frequency response also provides the magnitude and phase values for all steady-state step and sinusoidal responses. Furthermore, the Nyquist and Bode plots are used to analyze the frequency response of SISO LTI systems.

In the same way, the frequency response of MIMO LTI systems is the Fourier transform of the system, i.e. $[G(j\omega)]$, which provides the same information, albeit with potential coupling between all inputs and outputs via the transfer function matrix. However, recalling the definition of the singular value decomposition (SVD), one can model the transfer function matrix at a particular frequency as

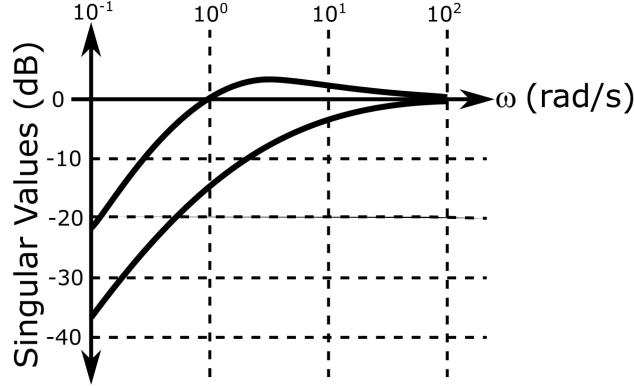
$$[G(j\omega)] = U(j\omega)\Sigma(j\omega)V^{-1}(j\omega) \quad (3.17)$$

where each of these matrices depends on frequency. Though the values of the unitary matrices, U and V , may change. The gain of the “directions” from input to output encoded in these unitary matrices is captured by the singular values within Σ . Thus, frequency response of MIMO LTI systems is typically done by plotting the singular values as a function of ω called the **singular value plot** also known as the **sigma plot** or **σ -plot**.

As an example, consider the system

$$[G(s)] = \begin{bmatrix} \frac{s}{s+3} & \frac{-6s}{s^2+6s+9} \\ 0 & \frac{s}{s+3} \end{bmatrix} \quad (3.18)$$

which provides the following σ -plot



Signal and System Norms

Consider a piecewise continuous signal vector, $\vec{u}(t) \in \mathbb{R}^{n_u}$. Then, one can define the **signal p -norm**, also known as the **signal \mathcal{L}_p -norm** as

$$\|\vec{u}(t)\|_p = \left(\int_0^\infty \sum_{i=1}^{n_u} |u_i(t)|^p dt \right)^{\frac{1}{p}} \quad (3.19)$$

where $u_i(t)$ is the i^{th} element of $\vec{u}(t)$, which may or may not be finite. Of particular interest is the **signal \mathcal{L}_1 -norm**, i.e.

$$\|\vec{u}(t)\|_1 = \int_0^\infty \sum_{i=1}^{n_u} |u_i(t)| dt \quad (3.20)$$

and the **signal \mathcal{L}_2 -norm**, i.e.

$$\|\vec{u}(t)\|_2 = \left(\int_0^\infty \sum_{i=1}^{n_u} u_i(t)^2 dt \right)^{\frac{1}{2}} \quad (3.21)$$

which can be redefined as

$$\|\vec{u}(t)\|_2 = \left(\int_0^\infty \text{Tr} [\vec{u}(t)^T \vec{u}(t)] dt \right)^{\frac{1}{2}} \quad (3.22)$$

or by Parseval's theorem

$$\|\vec{u}(t)\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Tr} [\vec{u}(j\omega)^* \vec{u}(j\omega)] d\omega \right)^{\frac{1}{2}} \quad (3.23)$$

Next, consider the LTI system $\vec{y}(s) = G(s)\vec{u}(s)$. Then, the $p \leftarrow q$ -induced system norms, also known as the $\mathcal{L}_{p,q}$ induced system norms, can be defined as

$$\|G\|_{p \leftarrow q} = \max_{0 \neq \|\vec{u}\|_q \leq \infty, \vec{x}(0)=0} \frac{\|\vec{y}\|_p}{\|\vec{u}\|_q} \quad (3.24)$$

which can also be stated as the smallest constant, c , such that $\|\vec{y}\|_p \leq c\|\vec{u}\|_q$. Note that if G is unstable, then $\|\vec{y}\|_p \rightarrow \infty$ and $\|G\|_{p \leftarrow q} = \infty$. Of particular interest is the $\|G\|_{2 \leftarrow 2}$ norms which is also known as the \mathcal{H}_∞ -norm of G for stable G , and is denoted, in this case by

$$\|G\|_\infty = \|G\|_{2 \leftarrow 2} \quad (3.25)$$

which is finite if and only if G is strictly proper, e.g. $D = 0$ in its LTI state-space model, with no poles on the $j\omega$ axis. By definition of the LTI system as $\vec{y}(j\omega) = G(j\omega)\vec{u}(j\omega)$, and the definition of the singular values of the transfer function matrix, $G(j\omega)$, one can write

$$\bar{\sigma}(G(j\omega)) = \max_{0 \neq \|\vec{u}(j\omega)\|_2} \frac{\|G(j\omega)\vec{u}(j\omega)\|_2}{\|\vec{u}(j\omega)\|_2} \quad (3.26)$$

where $\bar{\sigma}(G(j\omega))$ is the maximum singular value of the transfer function matrix $G(j\omega)$ evaluated at ω . Thus, over the entire frequency spectrum, one has

$$\|G\|_\infty = \max_{\omega} \bar{\sigma}(G(j\omega)) \quad (3.27)$$

which for SISO systems is simply the maximum magnitude of the Bode plot. For MIMO systems, the \mathcal{H}_∞ -norm is simply the maximum singular value across all frequencies, i.e. the peak on the σ -plot.

Lastly, the \mathcal{H}_2 -norm can be defined for a stable LTI system, G , as

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \|G(j\omega)\|_F^2 d\omega \right)^{\frac{1}{2}} \quad (3.28)$$

where $\|G(j\omega)\|_F$ is the Frobenius or entry-wise $L_{2,2}$ norm of the transfer function matrix evaluated at $j\omega$. This can be alternatively written as

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Tr}[G(j\omega)G(j\omega)^*] d\omega \right)^{\frac{1}{2}} \quad (3.29)$$

which is finite if and only if G is strictly proper, e.g. $D = 0$ in its state-space representation, with no poles on the $j\omega$ axis. It should also be noted that for SISO systems, the \mathcal{H}_2 -norm becomes

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} |G(j\omega)|^2 d\omega \right)^{\frac{1}{2}} \quad (3.30)$$

which is a measure of the area under the Bode plot of the system.

Using the SVD of $G(j\omega) = U(j\omega)\Sigma(j\omega)V(j\omega)^*$, one has

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Tr}[V(j\omega)\Sigma(j\omega)^*U(j\omega)^*U(j\omega)\Sigma(j\omega)V(j\omega)^*] d\omega \right)^{\frac{1}{2}} \quad (3.31)$$

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \text{Tr} [\Sigma(j\omega)^* \Sigma(j\omega)] d\omega \right)^{\frac{1}{2}} \quad (3.32)$$

$$\|G\|_2 = \left(\frac{1}{2\pi} \int_{-\infty}^{\infty} \sum_{i=1}^n \sigma_i(j\omega)^2 d\omega \right)^{\frac{1}{2}} \quad (3.33)$$

Thus, the \mathcal{H}_2 -norm can be considered as an average of the singular values also averaged across all frequencies.

An important interpretation of the \mathcal{H}_2 -norm occurs by considering $\vec{u}(t)$ as a white noise random process with unit variance, i.e.

$$\mathbb{E} [\vec{u}(t_1) \vec{u}(t_2)^T] = I_{n_u} \delta(t_1 - t_2) \quad (3.34)$$

Then, the \mathcal{H}_2 -norm squared is the expected power in the output signal, i.e.

$$\|G\|_2^2 = \mathbb{E} \left[\lim_{t_f \rightarrow \infty} \frac{1}{t_f} \int_0^{t_f} \text{Tr} [\vec{y}(t)^T \vec{y}(t)] dt \right] \quad (3.35)$$

which can be shown using the convolution integral for $\vec{y}(t)$. This provides a useful motivation for using \mathcal{H}_2 OCPs when *unobservable* random processes are significant inputs to the plant.

To calculate the \mathcal{H}_2 -norm, note that by Parseval's theorem one can write the transfer function matrix as the solution to the impulse response of the MIMO system with $D = 0$, i.e.

$$\|G\|_2 = \left(\int_0^{\infty} \text{Tr} [Ce^{At} BB^T e^{A^T t} C^T] dt \right)^{\frac{1}{2}} \quad (3.36)$$

$$\|G\|_2^2 = \text{Tr} \left[C \int_0^{\infty} e^{At} BB^T e^{A^T t} dt C^T \right] \quad (3.37)$$

and defining the **controllability gramian**, $W_C = W_C^T$, as

$$W_C = \int_0^{\infty} e^{At} BB^T e^{A^T t} dt \quad (3.38)$$

one has

$$\|G\|_2^2 = \text{Tr} [CW_C C^T] \quad (3.39)$$

where W_C is found by noting

$$\frac{d}{dt} e^{At} BB^T e^{A^T t} = Ae^{At} BB^T e^{A^T t} + e^{At} BB^T e^{A^T t} A^T \quad (3.40)$$

and integrating from 0 to ∞ , one has

$$e^{At} BB^T e^{A^T t} \Big|_0^{\infty} = \int_0^{\infty} Ae^{At} BB^T e^{A^T t} dt + \int_0^{\infty} e^{At} BB^T e^{A^T t} A^T dt \quad (3.41)$$

or

$$-BB^T = AW_C + W_C A^T \quad (3.42)$$

Thus, W_C is the unique solution to

$$AW_C + W_C A^T + BB^T = 0 \quad (3.43)$$

which exists if and only if A is stable.

Alternatively, one can also write

$$\|G\|_2 = \left(\int_0^\infty \text{Tr} [B^T e^{A^T t} C^T C e^{At} B] dt \right)^{\frac{1}{2}} \quad (3.44)$$

from which $W_O = W_O^T$ is the **observability gramian**

$$W_O = \int_0^\infty e^{A^T t} C^T C e^{At} dt \quad (3.45)$$

which can form the \mathcal{H}_2 -norm as

$$\|G\|_2^2 = \text{Tr} [B^T W_O B] \quad (3.46)$$

and is the unique solution to

$$A^T W_O + W_O A + C^T C = 0 \quad (3.47)$$

which exists if and only if A is stable.

Both Equations 3.43 and 3.47 are types of matrix Lyapunov equations and can be solved using numerical algorithms. A **matrix Lyapunov equation** is a type of LME of the form

$$A^T X + X A + Q = 0 \quad (3.48)$$

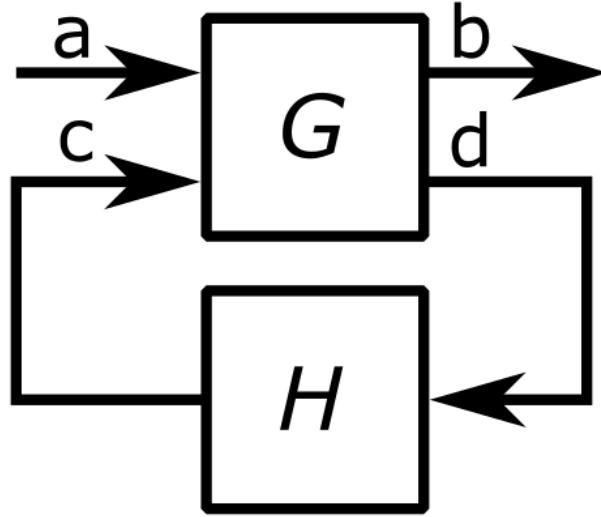
for which there is a unique solution

$$X = \int_0^\infty e^{A^T t} Q e^{At} dt \quad (3.49)$$

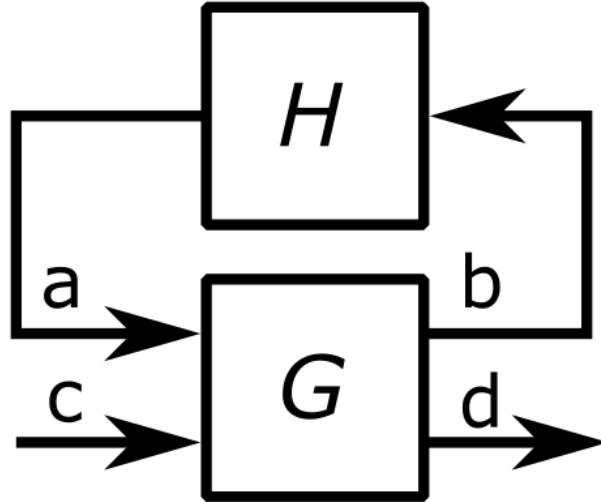
which is well-defined since A is assumed stable. X is typically obtained numerically using certain eigenvalue decompositions, not directly from this integral.

Linear Fractional Transformations

A **linear fractional transformations (LFT)** models a feedback connection of two matrices which can be used to represent MIMO LTI systems via the transfer function matrix. These can be either an **lower LFT**, i.e.



which is denoted as $F_L(G, H)$ or a **upper LFT**, i.e.



which is denoted as $F_U(G, H)$.

A lower LFT represents the following algebraic equations

$$\begin{bmatrix} \vec{b} \\ \vec{d} \end{bmatrix} = \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{c} \end{bmatrix} \quad (3.50)$$

$$\vec{c} = H\vec{d}$$

Substituting the third equation into the second equation provides

$$\vec{d} = G_{21}\vec{a} + G_{22}H\vec{d} \quad (3.51)$$

Thus, if $I_{n_d} - G_{22}H$ is non-singular, i.e. invertible, then the lower LFT is well-posed and this equation can be solved as

$$\vec{d} = (I_{n_d} - G_{22}H)^{-1} G_{21}\vec{a} \quad (3.52)$$

which can be substituted into the first equation with $\vec{c} = H\vec{d}$ to obtain

$$\begin{aligned} \vec{b} &= \left(G_{11} + G_{12}H(I_{n_d} - G_{22}H)^{-1}G_{21} \right) \vec{a} \\ \vec{b} &= F_L(G, H)\vec{a} \end{aligned} \quad (3.53)$$

An upper LFT represents the following algebraic equations

$$\begin{aligned} \begin{bmatrix} \vec{b} \\ \vec{d} \end{bmatrix} &= \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{c} \end{bmatrix} \\ \vec{a} &= H\vec{b} \end{aligned} \quad (3.54)$$

Substituting the third equation into the first equation provides

$$\vec{b} = G_{11}H\vec{b} + G_{12}\vec{c} \quad (3.55)$$

Thus, if $I_{n_b} - G_{11}H$ is non-singular, i.e. invertible, then the upper LFT is well-posed and this equation can be solved as

$$\vec{b} = (I_{n_b} - G_{11}H)^{-1} G_{12}\vec{c} \quad (3.56)$$

which can be substituted into the second equation with $\vec{a} = H\vec{b}$ to obtain

$$\begin{aligned} \vec{b} &= \left(G_{22} + G_{21}H(I_{n_b} - G_{11}H)^{-1}G_{12} \right) \vec{a} \\ \vec{d} &= F_U(G, H)\vec{c} \end{aligned} \quad (3.57)$$

An important property of LFTs is that one can represent block diagram algebra for serial, parallel, feedback, and inverse interconnections of LFTs as LFTs.

As an example of a serial and parallel interconnection, consider the two upper LFTs

$$\begin{aligned} \begin{bmatrix} \vec{b}_1 \\ \vec{d}_1 \end{bmatrix} &= \begin{bmatrix} L_{11} & L_{12} \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} \vec{a}_1 \\ \vec{c}_1 \end{bmatrix} \\ \vec{a}_1 &= H_1\vec{b}_1 \end{aligned} \quad (3.58)$$

and

$$\begin{aligned} \begin{bmatrix} \vec{b}_2 \\ \vec{d}_2 \end{bmatrix} &= \begin{bmatrix} N_{11} & N_{12} \\ N_{21} & N_{22} \end{bmatrix} \begin{bmatrix} \vec{a}_2 \\ \vec{c}_2 \end{bmatrix} \\ \vec{a}_2 &= H_2\vec{b}_2 \end{aligned} \quad (3.59)$$

connected in series from $F_U(L, H_1)$ to $F_U(N, H_2)$, i.e.

$$\vec{c}_2 = \vec{d}_1 \quad (3.60)$$

Combining, one has the upper LFT

$$\begin{aligned} \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vec{d}_2 \end{bmatrix} &= \begin{bmatrix} L_{11} & 0 & L_{12} \\ N_{12}L_{21} & N_{11} & N_{21}L_{22} \\ N_{22}L_{21} & N_{21} & N_{22}L_{22} \end{bmatrix} \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{c}_1 \end{bmatrix} \\ \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \end{bmatrix} &= \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix} \end{aligned} \quad (3.61)$$

or connected in parallel, i.e.

$$\vec{c} = \vec{c}_1 = \vec{c}_2 \quad \& \quad \vec{d} = \vec{d}_1 + \vec{d}_2 \quad (3.62)$$

Combining, one has the upper LFT

$$\begin{aligned} \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \\ \vec{d} \end{bmatrix} &= \begin{bmatrix} L_{11} & 0 & L_{12} \\ 0 & N_{11} & N_{12} \\ L_{21} & N_{21} & L_{22} + N_{22} \end{bmatrix} \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \\ \vec{c} \end{bmatrix} \\ \begin{bmatrix} \vec{a}_1 \\ \vec{a}_2 \end{bmatrix} &= \begin{bmatrix} H_1 & 0 \\ 0 & H_2 \end{bmatrix} \begin{bmatrix} \vec{b}_1 \\ \vec{b}_2 \end{bmatrix} \end{aligned} \quad (3.63)$$

where for both LFTs, the H_1 and H_2 matrices form a “structured” upper LFT, i.e. zero matrices exist on the off-diagonal block elements.

As an example of a feedback interconnection, consider the upper LFT

$$\begin{aligned} \begin{bmatrix} \vec{b} \\ \vec{d} \end{bmatrix} &= \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{c} \end{bmatrix} \\ \vec{d} &= H\vec{b} \end{aligned} \quad (3.64)$$

with the negative feedback interconnection for \vec{c} as

$$\vec{c} = \vec{r} - \vec{d} \quad (3.65)$$

where \vec{r} is some reference command input to the feedback loop. From the second equation, $\vec{d} = G_{21}\vec{a} + G_{22}\vec{c}$, assuming $I_{n_d} + G_{22}$ is invertible, one has by substituting for \vec{d}

$$\vec{c} = (I_{n_d} + G_{22})^{-1} (\vec{r} - G_{21}\vec{a}) \quad (3.66)$$

which can be substituted for \vec{c} to obtain the upper LFT

$$\begin{aligned} \begin{bmatrix} \vec{b} \\ \vec{d} \end{bmatrix} &= \begin{bmatrix} G_{11} - G_{12}(I_{n_d} + G_{22})^{-1}G_{21} & G_{12}(I_{n_d} + G_{22})^{-1} \\ (I_{n_d} + G_{22})^{-1}G_{21} & G_{22}(I_{n_d} + G_{22})^{-1} \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{r} \end{bmatrix} \\ \vec{a} &= H\vec{b} \end{aligned} \quad (3.67)$$

which is well-posed if $I_{n_b} - \left(G_{11} - G_{12} (I_{n_d} + G_{22})^{-1} G_{21} \right) H$ is invertible.

Consider the lower LFT

$$\begin{aligned} \begin{bmatrix} \vec{b} \\ \vec{d} \end{bmatrix} &= \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{c} \end{bmatrix} \\ \vec{c} &= H \vec{d} \end{aligned} \quad (3.68)$$

The **LFT inverse** can be obtained if G_{11} is invertible via the rewritten first equation

$$\vec{a} = G_{11}^{-1} \vec{b} - G_{11}^{-1} G_{12} \vec{c} \quad (3.69)$$

substituted into the second equation to obtain

$$\begin{aligned} \begin{bmatrix} \vec{a} \\ \vec{d} \end{bmatrix} &= \begin{bmatrix} G_{11}^{-1} & -G_{11}^{-1} G_{12} \\ G_{21} G_{11}^{-1} & G_{22} - G_{21} G_{11}^{-1} G_{12} \end{bmatrix} \begin{bmatrix} \vec{b} \\ \vec{c} \end{bmatrix} \\ \vec{c} &= H \vec{d} \end{aligned} \quad (3.70)$$

which is well-posed if $I_{n_d} - (G_{22} - G_{21} G_{11}^{-1} G_{12}) H$ is invertible.

Likewise, consider the upper LFT

$$\begin{aligned} \begin{bmatrix} \vec{b} \\ \vec{d} \end{bmatrix} &= \begin{bmatrix} G_{11} & G_{12} \\ G_{21} & G_{22} \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{c} \end{bmatrix} \\ \vec{a} &= H \vec{b} \end{aligned} \quad (3.71)$$

The **LFT inverse** can be obtained if G_{22} is invertible via the rewritten second equation

$$\vec{c} = -G_{22}^{-1} G_{21} \vec{a} + G_{22}^{-1} \vec{d} \quad (3.72)$$

substituted into the first equation to obtain

$$\begin{aligned} \begin{bmatrix} \vec{b} \\ \vec{c} \end{bmatrix} &= \begin{bmatrix} G_{11} - G_{12} G_{22}^{-1} G_{21} & G_{12} G_{22}^{-1} \\ -G_{22}^{-1} G_{21} & G_{22}^{-1} \end{bmatrix} \begin{bmatrix} \vec{a} \\ \vec{d} \end{bmatrix} \\ \vec{a} &= H \vec{b} \end{aligned} \quad (3.73)$$

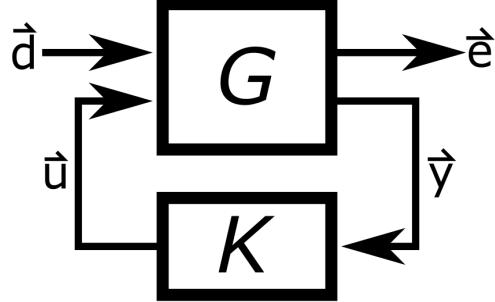
which is well-posed if $I_{n_b} - (G_{11} - G_{12} G_{22}^{-1} G_{21}) H$ is invertible.

3.2 MIMO LTI Feedback Control Systems

When one can use MIMO LTI control systems, one may use a combination of feedforward terms, i.e. control gains dependent on \vec{r} , tracking error terms, i.e. control gains dependent on \vec{e} , output terms, i.e. \vec{y} , other terms, e.g. integrated signals and derivative signals as well as coupling between all of these. Thus, for studying MIMO LTI systems, it is useful to define a generalized framework to assess system properties including stability, controllability, observability, and robustness.

Generalized LTI Feedback Control System

Consider the generalized LTI feedback control system architecture



which is denoted by $F_L(G, K)$ and is a linear fractional transformation (LFT). The vector input, $\vec{d} \in \mathbb{R}^{n_d}$, to the generalized plant is known as the **generalized disturbance** which typically includes reference commands, \vec{r} , process noise, \vec{w} , and measurement noise, \vec{v} . The vector output, $\vec{e} \in \mathbb{R}^{n_e}$, from the generalized plant is known as the **generalized error** which typically includes at least the weighted tracking error and the weighted control effort. The other vectors are the **generalized control input** vector $\vec{u} \in \mathbb{R}^{n_u}$ and the **generalized output**, $\vec{y} \in \mathbb{R}^{n_y}$.

In this context, the **generalized plant**, G , contains the original plant dynamics, actuators, weighting filters, and signal routing and operations on the disturbance, error, state, inputs, and output signals. Thus, the generalized output may be the tracking error instead of the “true” plant output or it may contain the reference command and the plant output. Using LFTs, the LTI state-space model for the generalized plant, G , can be written as

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + [B_1 \quad B_2] \begin{bmatrix} \vec{d}(t) \\ \vec{u}(t) \end{bmatrix} \\ \begin{bmatrix} \vec{e}(t) \\ \vec{y}(t) \end{bmatrix} &= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \vec{d}(t) \\ \vec{u}(t) \end{bmatrix}\end{aligned}\tag{3.74}$$

with the **generalized LTI feedback controller**, K , designed as

$$\begin{aligned}\dot{\vec{x}}_K &= A_K \vec{x}_K + B_K \vec{y} \\ \vec{u} &= C_K \vec{x}_K + D_K \vec{y}\end{aligned}\tag{3.75}$$

Combining, one has for the input and output vectors

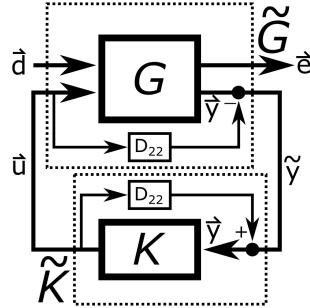
$$\begin{bmatrix} I & -D_K \\ -D_{22} & I \end{bmatrix} \begin{bmatrix} \vec{u}(t) \\ \vec{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & C_K \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + \begin{bmatrix} 0 \\ D_{21} \end{bmatrix} [\vec{d}(t)]\tag{3.76}$$

thus, the interconnection of G and K is well-posed if and only if $(I - D_{22}D_K)^{-1}$ exists, i.e. \vec{u}, \vec{y} can be substituted into the state equation

$$\begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\vec{x}}_K(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_K \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \vec{d}(t) + \begin{bmatrix} B_2 & 0 \\ 0 & B_K \end{bmatrix} \begin{bmatrix} \vec{u}(t) \\ \vec{y}(t) \end{bmatrix}\tag{3.77}$$

and a unique solution can be found.

Note that in control synthesis, D_{22} is set to 0 without loss of generality as one can alternatively use **loop-shifting** which can be visualized as the following alteration to $F_L(G, K)$ as



to form $F_L(\tilde{G}, \tilde{K})$ from \tilde{G} and \tilde{K} and then undo the loop-shifting to get the original K . Thus, one typically assumes the generalized LTI plant, G , can be specified as

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + [B_1 \quad B_2] \begin{bmatrix} \vec{d}(t) \\ \vec{u}(t) \end{bmatrix} \\ \begin{bmatrix} \vec{e}(t) \\ \vec{y}(t) \end{bmatrix} &= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & 0 \end{bmatrix} \begin{bmatrix} \vec{d}(t) \\ \vec{u}(t) \end{bmatrix}\end{aligned}\quad (3.78)$$

In this case, the state equation for the closed-loop system can be rewritten as

$$\begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\vec{x}}_K(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ 0 & A_K \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ 0 \end{bmatrix} \vec{d}(t) + \begin{bmatrix} B_2 & 0 \\ 0 & B_K \end{bmatrix} \begin{bmatrix} \vec{u}(t) \\ \vec{y}(t) \end{bmatrix} \quad (3.79)$$

with the following relationship for $\vec{u}(t)$ and $\vec{y}(t)$

$$\begin{bmatrix} I & -D_K \\ 0 & I \end{bmatrix} \begin{bmatrix} \vec{u}(t) \\ \vec{y}(t) \end{bmatrix} = \begin{bmatrix} 0 & C_K \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + \begin{bmatrix} 0 \\ D_{21} \end{bmatrix} \begin{bmatrix} \vec{d}(t) \end{bmatrix} \quad (3.80)$$

$$\begin{bmatrix} \vec{u}(t) \\ \vec{y}(t) \end{bmatrix} = \begin{bmatrix} I & D_K \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 & C_K \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + \begin{bmatrix} I & D_K \\ 0 & I \end{bmatrix} \begin{bmatrix} 0 \\ D_{21} \end{bmatrix} \begin{bmatrix} \vec{d}(t) \end{bmatrix} \quad (3.81)$$

$$\begin{bmatrix} \vec{u}(t) \\ \vec{y}(t) \end{bmatrix} = \begin{bmatrix} D_K C_2 & C_K \\ C_2 & 0 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + \begin{bmatrix} D_K D_{21} \\ D_{21} \end{bmatrix} \begin{bmatrix} \vec{d}(t) \end{bmatrix} \quad (3.82)$$

Then, by substitution, one can obtain the closed-loop LTI state-space system model as

$$\begin{aligned}\begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\vec{x}}_K(t) \end{bmatrix} &= \begin{bmatrix} A + B_2 D_K C_2 & B_2 C_K \\ B_K C_2 & A_K \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + \begin{bmatrix} B_1 + B_2 D_K D_{21} \\ B_K D_{21} \end{bmatrix} \vec{d}(t) \\ \vec{e}(t) &= [C_1 + D_{12} D_K C_2 \quad D_{12} C_K] \begin{bmatrix} \vec{x}(t) \\ \vec{x}_K(t) \end{bmatrix} + (D_{11} + D_{12} D_K D_{21}) \vec{d}(t)\end{aligned}\quad (3.83)$$

with the definitions for the closed-loop state matrix, A_L , as

$$A_L = \begin{bmatrix} A + B_2 D_K C_2 & B_2 C_K \\ B_K C_2 & A_K \end{bmatrix} \quad (3.84)$$

the closed-loop input matrix, B_L , as

$$B_L = \begin{bmatrix} B_1 + B_2 D_K D_{21} \\ B_K D_{21} \end{bmatrix} \quad (3.85)$$

the closed-loop output matrix, C_L , as

$$C_L = [C_1 + D_{12} D_K C_2 \quad D_{12} C_K] \quad (3.86)$$

and the closed-loop feedthrough matrix, D_L , as

$$D_L = D_{11} + D_{12} D_K D_{21} \quad (3.87)$$

which for stability of the generalized feedback control system, one requires that A_L is stable.

State and Output Feedback Control

If $A_K = C_K = B_K = 0$, one has **static-controller feedback control**, a type of **fixed-gain controller**. The general case is also known as a.k.a. **output feedback control**, where one has an output feedback controller as

$$\vec{u}(t) = D_K \vec{y}(t) \quad (3.88)$$

which results in a simplified closed-loop LTI state-space model

$$\begin{aligned} [\dot{\vec{x}}(t)] &= A_L \vec{x}(t) + B_L \vec{d}(t) \\ \vec{e}(t) &= C_L \vec{x}(t) + D_L \vec{d}(t) \end{aligned} \quad (3.89)$$

which results in a closed-loop state matrix, A_L , as

$$A_L = A + B_2 D_K C_{12} \quad (3.90)$$

a closed-loop input matrix, B_L , as

$$B_L = B_1 + B_2 D_K D_{21} \quad (3.91)$$

a closed-loop output matrix, C_L , as

$$C_L = C_1 + D_{12} D_K C_2 \quad (3.92)$$

and a closed-loop feedthrough matrix, D_L , as

$$D_L = D_{11} + D_{12} D_K D_{21} \quad (3.93)$$

Thus, an output feedback control system is stable if and only if $A + B_2 D_K C_{12}$ is stable.

A special case of fixed-gain feedback control is **state feedback control**, i.e. $C_2 = I$, $D_{21} = 0$, and state feedback controller

$$\vec{u}(t) = D_K \vec{x}(t) \quad (3.94)$$

with closed-loop state matrix, A_L , is

$$A_L = A + B_2 D_K \quad (3.95)$$

the closed-loop input matrix, B_L , is

$$B_L = B_1 \quad (3.96)$$

the closed-loop output matrix, C_L , as

$$C_L = C_1 + D_{12} D_K \quad (3.97)$$

and the closed-loop feedthrough matrix, D_L , as

$$D_L = D_{11} \quad (3.98)$$

Thus, a state feedback control system is stable if and only if $A + B_2 D_K$ is stable.

Observer Feedback Control

The general case of dynamic-controller feedback control is also known as **observer feedback control**, one defines the controller state as the **state estimate**, \hat{x} , i.e.

$$\vec{x}_K = \hat{x} \quad (3.99)$$

where an observer is designed to form \hat{x} and uses this to form the control input as

$$\vec{u}(t) = -K \hat{x}(t) \quad (3.100)$$

Here, an **open-loop observer** could be formed based on the linear state-space model for continuous-time, assuming the disturbances \vec{d} are unknown, as

$$\dot{\hat{x}} = A \hat{x} + B_2 \vec{u} \quad (3.101)$$

However, for feedback control, one receives the output signal from the output equation, and forms a **closed-loop observer** for continuous-time as

$$\begin{aligned} \dot{\hat{x}}(t) &= A \hat{x}(t) + B_2 \vec{u}(t) + L \left(\vec{y}(t) - \hat{y}(t) \right) \\ \hat{y}(t) &= C_2 \hat{x}(t) \end{aligned} \quad (3.102)$$

where \hat{y} is the output estimate based on the output equation model and L is the **Luenberger observer matrix**. Thus, with $\vec{u}(t) = -K \hat{x}(t)$, one can form the continuous-time **observer feedback control system** as the generalized LTI feedback controller

$$\begin{aligned} \dot{\hat{x}}(t) &= (A - B_2 K - LC_2) \hat{x}(t) + L \vec{y}(t) \\ \vec{u}(t) &= -K \hat{x}(t) \end{aligned} \quad (3.103)$$

where

$$A_K = A - B_2 K - LC_2 \quad (3.104)$$

$$B_K = L \quad (3.105)$$

$$C_K = -K \quad (3.106)$$

$$D_K = 0 \quad (3.107)$$

Thus, one has the closed-loop dynamics

$$\begin{aligned} \begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\hat{x}}(t) \end{bmatrix} &= \begin{bmatrix} A & -B_2K \\ LC_2 & A - B_2K - LC_2 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \hat{x}(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ LD_{21} \end{bmatrix} \vec{d}(t) \\ \vec{e}(t) &= [C_1 \ -D_{12}K] \begin{bmatrix} \vec{x}(t) \\ \hat{x}(t) \end{bmatrix} + D_{11} \vec{d}(t) \end{aligned} \quad (3.108)$$

However, to better assess the stability of the closed-loop system, consider the **state error**, $\vec{e}_x(t)$, defined as

$$\vec{e}_x(t) = \vec{x}(t) - \hat{x}(t) \quad (3.109)$$

Then, one has

$$\begin{aligned} \begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\vec{x}}(t) - \dot{\vec{e}}_x(t) \end{bmatrix} &= \begin{bmatrix} A & -B_2K \\ LC_2 & A - B_2K - LC_2 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}(t) - \vec{e}_x(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ LD_{21} \end{bmatrix} \vec{d}(t) \\ \vec{e}(t) &= [C_1 \ -D_{12}K] \begin{bmatrix} \vec{x}(t) \\ \vec{x}(t) - \vec{e}_x(t) \end{bmatrix} + D_{11} \vec{d}(t) \end{aligned} \quad (3.110)$$

which can be rearranged as

$$\begin{aligned} \begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\vec{x}}(t) - \dot{\vec{e}}_x(t) \end{bmatrix} &= \begin{bmatrix} (A - B_2K) & B_2K \\ (A - B_2K) & -A + B_2K + LC_2 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{e}_x(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ LD_{21} \end{bmatrix} \vec{d}(t) \\ \vec{e}(t) &= [(C_1 - D_{12}K) \ D_{12}K] \begin{bmatrix} \vec{x}(t) \\ \vec{e}_x(t) \end{bmatrix} + D_{11} \vec{d}(t) \end{aligned} \quad (3.111)$$

And subtracting the second row of the state equation from the first row, one has

$$\begin{aligned} \begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\vec{e}}_x(t) \end{bmatrix} &= \begin{bmatrix} (A - B_2K) & B_2K \\ 0 & A - LC_2 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{e}_x(t) \end{bmatrix} + \begin{bmatrix} B_1 \\ B_1 - LD_{21} \end{bmatrix} \vec{d}(t) \\ \vec{e}(t) &= [(C_1 - D_{12}K) \ D_{12}K] \begin{bmatrix} \vec{x}(t) \\ \vec{e}_x(t) \end{bmatrix} + D_{11} \vec{d}(t) \end{aligned} \quad (3.112)$$

Since the eigenvalues of an upper triangular matrix are dependent only on the block diagonal terms, an observer feedback control system is stable if and only if $A - B_2K$ and $A - LC_2$ are stable. This fact of linear observer feedback control systems having independent criteria for the design of L and K is known as the **separation principle**, a foundational result in control and perception theory which justifies the separate use of control and perception algorithms in dynamical systems applications. although it only technically applies to linear systems. Thus, this part of the textbook focuses on LTI state feedback control design. The subsequent parts of the textbook consider optimal state estimation and its application to perception of a flight vehicle's "state."

With this analysis in mind, a straightforward linear controls design method is **eigenvalue placement**, also known as **pole placement** where one “places” the eigenvalues of the linear feedback control system in the complex plane by choosing the feedback gain matrix appropriately, thereby setting the modal characteristics for the system response. This method can be considered as a generalization of the root locus technique for selecting a single gain parameter in a SISO LTI control system to selecting two gain matrices in a MIMO LTI control systems and its effect on the system’s root/poles/eigenvalues for control and state estimation. It should be noted that there are computer algorithms to solve the eigenvalue placement problem, e.g. `place` in MATLAB, but derivations for these generic algorithms are beyond the scope of this textbook. However, for MIMO LTI systems, one may not be able to set the eigenvalues arbitrarily as the eigenvalue equations also depend on the generalized plant state, input, and potentially the output matrices. The ability to arbitrarily set the eigenvalues leads to the concepts of controllability and observability for LTI systems. To place the poles of state feedback, the system must be state controllable. To place the poles of output feedback, the system must be output controllable. To place the poles of observer feedback, the system must be state controllable to place $A - B_2 K$ and observable to place $A - LC_2$.

Controllability and Observability of Continuous-Time LTI Systems

State controllability is defined as the ability to control a system to *any* state using some finite input. Related to this is the concept of **state reachability** and can be stated as follows. For continuous-time dynamical systems, a state \vec{x}' is reachable if for every finite $T > 0$, there exists an input function $u(t)$ with $0 < t \leq T$ such that the state goes from $x(0) = 0 \rightarrow x(T) = \vec{x}'$. Thus, reachability is generally a slightly weaker notion than controllability. However, for linear state-space systems, the sequence for reaching any state can be inverted to return to zero from any initial conditions, thus state reachability is equivalent to state controllability. Using linear algebra, one can look at the conditions for state reachability which will imply state controllability for continuous-time LTI state-space systems.

Consider the state equation for a continuous-time LTI state-space system

$$\dot{\vec{x}}(t) = A\vec{x}(t) + B\vec{u}(t) \quad (3.113)$$

Given initial state $\vec{x}(0) = 0$, the general state-space solution is

$$\vec{x}(t) = \int_0^t e^{A(t-\tau)} B\vec{u}(\tau) d\tau \quad (3.114)$$

using a change of variables of $\tau_2 = t - \tau$

$$\vec{x}(t) = \int_0^t -e^{A(\tau_2)} B\vec{u}(t - \tau_2) d\tau_2 \quad (3.115)$$

and using the **Cayley-Hamilton definition** of the matrix exponential, i.e.

$$e^{At} = \sum_{i=0}^{n-1} A^i \alpha_i(t) \quad (3.116)$$

where $\alpha_0, \dots, \alpha_{n-1}$ are coefficients which depend on A , one has

$$\vec{x}(t) = \int_0^t \sum_{i=0}^{n-1} -A^i \alpha_i(\tau_2) B\vec{u}(t - \tau_2) d\tau_2 \quad (3.117)$$

which can be rearranged using the properties of integrals and sums

$$\vec{x}(t) = \sum_{i=0}^{n-1} A^i B \int_0^t -\alpha_i(\tau_2) \vec{u}(t-\tau) d\tau_2 \quad (3.118)$$

Then, by letting $\beta_i(t) = \int_0^t -\alpha_i(\tau_2) \vec{u}(t-\tau) d\tau_2$, one has

$$\vec{x}(t) = \sum_{i=0}^{n-1} A^i B \beta_i(t) \quad (3.119)$$

and the sum can be written out explicitly as

$$\vec{x}(t) = B\beta_0(t) + AB\beta_1(t) + \dots + A^{n-1}B\beta_{n-1}(t) \quad (3.120)$$

which can also be separated into the product of two matrices

$$\vec{x}(t) = [B \ AB \ \dots \ A^{n-1}B] \begin{bmatrix} \beta_0(t) \\ \beta_1(t) \\ \vdots \\ \beta_{n-1}(t) \end{bmatrix} \quad (3.121)$$

Finally, since any control input $u(t)$ is allowed, any $\begin{bmatrix} \beta_0(t) \\ \beta_1(t) \\ \vdots \\ \beta_{n-1}(t) \end{bmatrix}$ can be constructed, thus any $\vec{x}(t)$ can be reached and the pair (A, B) is controllable if and only if **controllability matrix**, $[B \ AB \ \dots \ A^{n-1}B]$, is invertible, i.e.

$$\text{rank}([B \ AB \ \dots \ A^{n-1}B]) = n \quad (3.122)$$

A second test is the **Popov-Belevitch-Hautus (PBH) controllability test** which states (A, B) is controllable if and only if for all eigenvalues $\lambda \in \mathbb{C}$ of A , one has

$$\text{rank}([\lambda I - A \ B]) = n \quad (3.123)$$

A weaker notion than controllability is **stabilizability**. A system is said to be stabilizable if all *uncontrollable* state variables can be controlled. Thus, (A, B) is stabilizable if and only if for all eigenvalues $\lambda \in \mathbb{C}$ of A with $\text{Real}(\lambda) \geq 0$, one has

$$\text{rank}([\lambda I - A \ B]) = n \quad (3.124)$$

Lastly, a third test uses the **controllability Gramian**, which for continuous-time can be written as

$$W_C(t) = \int_0^t e^{A\tau} BB^T e^{A\tau} d\tau \quad (3.125)$$

It can be shown that if and only if $W_C(t)$ is positive definite for *any* $t > 0$, then the system is controllable. The equation above can be reduced to solving the equation for continuous-time as

$$AW_C + W_C A^T = -BB^T \quad (3.126)$$

then, checking if $W_C > 0$. Thus, the eigenvalues of the $n \times n$ Gramian, W_C , characterize the relative degree of controllability.

In addition to state controllability, there are two other common notions of controllability. The first is **output controllability** which is the ability of an input to move the *output* from any initial condition to any final condition in finite time. This type of analysis naturally involves the output matrix in addition to the input matrix, e.g. the continuous-time **output controllability matrix** can be shown to be

$$\begin{bmatrix} C & CAB & CA^2B & \cdots & CA^{n_x-1}B & D \end{bmatrix} \quad (3.127)$$

It is also important to point out that state and output controllability are not equivalent nor does one imply the other. The second form is **controllability under constraints** which may be imposed upon practical systems modeled as an LTI system. Such constraints may be inherent to the system, e.g. saturating actuator, or imposed by the control designer, e.g. due to safety-related concerns. The effect of constraints to a system is a vast larger topic in control and is mentioned later in this textbook.

Observability for Continuous-Time LTI Systems

Observability is defined as the ability to observe system's *past* initial state, $\vec{x}(0)$, i.e. if, for some finite time interval, $[0, t_f]$, inputs $\vec{u}(t)$, and outputs $\vec{y}(t)$, the initial state $\vec{x}(0)$ can be determined. Consider the $n - 1$ continuous output derivatives which necessitate measurements of $y(t)$ over the time interval, $[0, t_f]$

$$\begin{aligned} \vec{y}(0) &= C\vec{x}_0 \\ \dot{\vec{y}}(0) &= C\dot{\vec{x}}(0) = C(A\vec{x}(0) + B\vec{u}(0)) \\ \ddot{\vec{y}}(0) &= C\ddot{\vec{x}}(0) = C\frac{d}{dt}(A\vec{x}(0) + B\vec{u}(0)) \\ \ddot{\vec{y}}(0) &= CA(A\vec{x}(0) + B\vec{u}(0)) + CB\dot{\vec{u}}(0) \\ &\vdots = \vdots \\ \vec{y}^{[n-1]}(0) &= CA^{n-1}\vec{x}(0) + CA^{n-2}B\vec{u}^{[n-2]}(0) + \cdots + CB\dot{\vec{u}}(0) \end{aligned} \quad (3.128)$$

Thus, as $\vec{y}(0)$ and $\vec{u}(0)$ and all their derivatives are known, any $\vec{x}(0)$ can be estimated and the pair (A, C)

is observable if and only if the **observability matrix**, $\begin{bmatrix} C \\ CA \\ \dots \\ CA^{n-1} \end{bmatrix}$, is invertible, i.e.

$$\text{rank} \left(\begin{bmatrix} C \\ CA \\ \dots \\ CA^{n-1} \end{bmatrix} \right) = n \quad (3.129)$$

A second test is the **Popov-Belevitch-Hautus (PBH) observability test** which states (A, C) is observable if and only if for all eigenvalues $\lambda \in \mathbb{C}$ of A , one has

$$\text{rank} ([\lambda I - A \quad C]) = n \quad (3.130)$$

A weaker notion than observability is **detectability**. A system is said to be detectable if all *uncontrollable* state variables can be observed. Thus, (A, C) is detectable if and only if for all eigenvalues $\lambda \in \mathbb{C}$ of A with $\text{Real}(\lambda) \geq 0$, one has

$$\text{rank}([\lambda I - A \quad C]) = n \quad (3.131)$$

Lastly, a third method uses the **observability Gramian** to determine the observability for continuous-time LTI state-space systems as

$$W_O(t) = \int_0^t e^{A\tau} CC^T e^{A\tau} d\tau \quad (3.132)$$

It can be shown that if and only if $W_O(t)$ is positive definite for *any* $t > 0$, then the system is observable. The Gramian can be shown to be the solution in continuous-time for the equation

$$AW_O + W_O A^T = -CC^T \quad (3.133)$$

then, checking if $W > 0$. Thus, the eigenvalues of the $n \times n$ Gramian, W_O characterize the relative degree of observability.

3.3 Robust Stability of LTI Systems

Stability robustness to model uncertainties is a crucial design criterion for the feedback control systems of flight vehicles. Historically, the most widely used measure of stability robustness in flight vehicles has been single-loop gain and phase stability margins for SISO LTI systems, typically designated as at least ± 6 dB gain margin and 45° phase margin. However, more advanced methods exist for stability robustness analyses of LTI systems due to various types of uncertainties which try to determine bounds on how large the uncertainties can be before the LTI system becomes unstable. This section discusses general types of uncertainties, generalized framework for studying uncertain LTI systems, and primary robust stability results using this framework.

Types of Uncertainties in Dynamical Systems

Fundamentally, dynamical systems are models of real world processes which are typically not completely understood and thus have some uncertainty. Thus, to account for model uncertainty formally, one can define different types of model uncertainty, either as parametric uncertainty or dynamic uncertainty.

Parametric uncertainty is specified as parameters which are unknown, but are members of some set of values called an **uncertainty set**. For dynamical systems, these parameters can be **complex parametric uncertainty** or **real parametric uncertainty** where either of these types can be scalar, i.e. some $\alpha \in \mathbb{C}$ or some $\beta \in \mathbb{R}$, or matrix-defined sets. A *probabilistic analysis approach* to modeling parametric uncertainty would be to specify the parameters as random variables distributed according to a distribution function where this approach also allows one to model dependencies between parameters. An alternative *worst-case analysis approach* is to use equally likely values within some finite uncertainty set. As probability theory is beyond the scope of this part of the textbook, robustness analysis of LTI systems under parametric uncertainty will use the worst-case analysis approach.

An important type of uncertainty set for complex parameters is the uncertainty disk defined as

$$\alpha \in \text{Disk} \left(\frac{1-m}{1+m}, \frac{1+m}{1-m} \right) = \left\{ \alpha = \frac{1-m\delta}{1+m\delta} \in \mathbb{C} : \delta \in \mathbb{C}, |\delta| \leq 1 \right\} \quad (3.134)$$

where $m \in [0, 1)$, the radius of the disk is

$$\frac{1}{2} \left(\frac{1+m}{1-m} - \frac{1-m}{1+m} \right) = \frac{1}{2} \left(\frac{1+2m+m^2 - 1+2m-m^2}{(1+m)(1-m)} \right) = \frac{2m}{(1+m)(1-m)} \quad (3.135)$$

and the center of the disk is

$$\frac{1-m}{1+m} + \frac{2m}{(1+m)(1-m)} = \frac{1-2m+m^2+2m}{(1+m)(1-m)} = m/2 \quad (3.136)$$

which can also be used to define *Disk* instead of m . Notably, as $m \rightarrow 1$, this disk converges to the right half of the complex plane (RHP). An important type of uncertainty set for real parameters is the closed interval defined as

$$\beta \in [\underline{\beta}, \bar{\beta}] = \left\{ \beta = \frac{\bar{\beta}+\beta}{2} \delta_\beta^2 + \frac{\bar{\beta}-\beta}{2} \delta_\beta + \beta_0 : \delta_\beta \in [-1, 1] \right\} \quad (3.137)$$

where δ is the normalized uncertainty. Notably, $\delta_\beta = 0$ corresponds to the nominal β_0 , $\delta_\beta = -1$ corresponds to $\underline{\beta}$ and $\delta_\beta = 1$ corresponds to $\bar{\beta}$.

Dynamic uncertainty which can be **LTI dynamic uncertainty** or **nonlinear dynamic uncertainty**. Notably, one can relate LTI dynamic uncertainty to complex parametric uncertainty. This textbook will focus on LTI dynamic uncertainty for robustness analysis of LTI systems as robustness analysis for nonlinear and/or time-varying dynamic uncertainty requires more general **integral quadratic constraints** which will not be covered in this textbook.

An important uncertainty set for LTI dynamic uncertainty is defined as

$$G(s) \in \mathcal{M}_W = \{G(s) = G_0(s)(I + W(s)\Delta(s)) : \Delta(s) \text{ LTI}, \|\Delta\|_\infty \leq 1\} \quad (3.138)$$

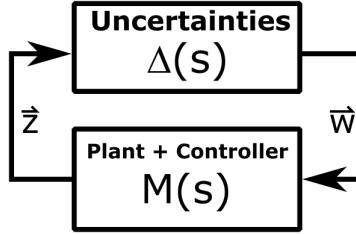
where $G_0(s)$ is the nominal LTI model, $W(s)$ is an LTI weight, and $\Delta(s)$ is the normalized uncertainty. Noticeably any $G(s) \in \mathcal{M}_W$ satisfies

$$\left| \frac{G(j\omega) - G_0(j\omega)}{G_0(j\omega)} \right| \leq |W(j\omega)| \quad \forall \omega \quad (3.139)$$

Thus, $W(j\omega)$ is a bound on the relative error at all frequencies from $G_0(j\omega)$. Furthermore, as the relative error bound is unaffected by the phase of $W(s)$, $W(s)$ can be chosen without loss of generality to be stable and minimum phase and modeled as a transfer function or state-space model. Lastly, if Δ is assumed to be stable, then one is restricting that $G(s)$ can only have RHP poles at the same locations as $G_0(s)$ which is often used in robustness analysis.

Generalized LTI Uncertainty Model

Consider the generalized LTI uncertainty model architecture



which is denoted by $F_U(M, \Delta)$ and is a linear fractional transformation (LFT), i.e.

$$\begin{bmatrix} \vec{z} \\ \vec{d} \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \vec{w} \\ \vec{e} \end{bmatrix}$$

$$\vec{w} = \Delta \vec{z} \quad (3.140)$$

Here the generalized uncertainty set, $\Delta(s)$ with $\|\Delta\|_\infty \leq 1$, contains all parametric and stable LTI dynamic uncertainties in the nominal stable LTI system, $M(s)$, e.g. the nominal generalized LTI plant and controller. Thus, the nominal stable LTI system dynamics correspond to $\Delta = 0$ and this LFT is well-posed if $I_{n_z} - M_{11}(\infty)\Delta(\infty)$ is invertible. Recalling the nature of LFTs, Δ may have a “structure” depending on the types of uncertainties, e.g. multiple scalar uncertainties and/or real parametric uncertainties. However, if Δ is a single complex uncertainty, i.e. $\Delta \in \mathbb{C}^{n_w \times n_z}$ or a single LTI dynamic uncertainty, then Δ is an **unstructured uncertainty set**.

Using LFTs, the LTI state-space model for the nominal model, M , can be written as

$$\begin{aligned} \dot{\vec{x}}(t) &= A \vec{x}(t) + [B_1 \quad B_2] \begin{bmatrix} \vec{w}(t) \\ \vec{d}(t) \end{bmatrix} \\ \begin{bmatrix} \vec{z}(t) \\ \vec{e}(t) \end{bmatrix} &= \begin{bmatrix} C_1 \\ C_2 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix} \begin{bmatrix} \vec{w}(t) \\ \vec{d}(t) \end{bmatrix} \end{aligned} \quad (3.141)$$

with the generalized uncertainty set, Δ , written as

$$\begin{aligned} \dot{\vec{x}}_\Delta &= A_\Delta \vec{x}_\Delta + B_\Delta \vec{z} \\ \vec{w} &= C_\Delta \vec{x}_\Delta + D_\Delta \vec{z} \end{aligned} \quad (3.142)$$

where if Δ contains only real or complex uncertainty, then $A_\Delta = B_\Delta = C_\Delta = 0$ with no state vector, \vec{x}_Δ .

Combining, one has for the uncertainty input and output vectors

$$\begin{bmatrix} I & -D_\Delta \\ -D_{11} & I \end{bmatrix} \begin{bmatrix} \vec{w}(t) \\ \vec{z}(t) \end{bmatrix} = \begin{bmatrix} 0 & C_\Delta \\ C_1 & 0 \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_\Delta(t) \end{bmatrix} + \begin{bmatrix} 0 \\ D_{12} \end{bmatrix} \begin{bmatrix} \vec{d}(t) \end{bmatrix} \quad (3.143)$$

thus, the interconnection of M and Δ is well-posed if and only if $I_{n_z} - D_{11}D_\Delta$ is invertible. Thus, similar to the generalized LFT feedback control system using loop-shifting, one can form the closed-loop LTI state-space system model as

$$\begin{aligned} \begin{bmatrix} \dot{\vec{x}}(t) \\ \dot{\vec{x}}_\Delta(t) \end{bmatrix} &= \begin{bmatrix} A + B_1 D_\Delta C_1 & B_1 C_\Delta \\ B_\Delta C_1 & A_\Delta \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_\Delta(t) \end{bmatrix} + \begin{bmatrix} B_2 + B_1 D_\Delta D_{21} \\ B_\Delta D_{12} \end{bmatrix} \vec{d}(t) \\ \vec{e}(t) &= \begin{bmatrix} C_2 + D_{21} D_\Delta C_1 & D_{21} C_\Delta \end{bmatrix} \begin{bmatrix} \vec{x}(t) \\ \vec{x}_\Delta(t) \end{bmatrix} + (D_{22} + D_{21} D_\Delta D_{12}) \vec{d}(t) \end{aligned} \quad (3.144)$$

with the definitions for the closed-loop state matrix, A_L , as

$$A_L = \begin{bmatrix} A + B_1 D_\Delta C_1 & B_1 C_\Delta \\ B_\Delta C_2 & A_\Delta \end{bmatrix} \quad (3.145)$$

the closed-loop input matrix, B_L , as

$$B_L = \begin{bmatrix} B_2 + B_1 D_\Delta D_{12} \\ B_\Delta D_{12} \end{bmatrix} \quad (3.146)$$

the closed-loop output matrix, C_L , as

$$C_L = [C_2 + D_{21} D_\Delta C_1 \quad D_{21} C_\Delta] \quad (3.147)$$

and the closed-loop feedthrough matrix, D_L , as

$$D_L = D_{22} + D_{21} D_\Delta D_{12} \quad (3.148)$$

which for stability of the generalized LTI uncertainty system, one requires that A_L is stable.

However, for stability robustness analysis, if one assumes M and Δ are stable and $F_U(M, \Delta)$ is well-posed, then $F_U(M, \Delta)$ has a pole at s_p if and only if $\det(s_p I - A_L) = 0$ which can be shown through determinant formulas and M and Δ stability. This only occurs if and only if $\det(I - M_{11}(s_p)\Delta(s_p)) = 0$. This implies that $F_U(M, \Delta)$ is unstable if and only if $\det(I - M_{11}(s_p)\Delta(s_p)) \neq 0$ for some s_p in the closed RHP. This can also be seen from the LFT from d to e as

$$F_U(M, \Delta) = M_{22} + M_{21}\Delta(I - M_{11}\Delta)^{-1}M_{12} \quad (3.149)$$

where Δ , M_{11} , M_{12} , M_{21} , and M_{22} are all stable by assumption. Hence, $F_U(M, \Delta)$ is stable if and only if $(I - M_{11}\Delta)^{-1}$ is unstable.

This allows the definition that the generalized LTI uncertainty system, $F_U(M, \Delta)$, is **robustly stable** if and only if $I - M_{11}\Delta$ is invertible for all $\Delta \in \Delta$ where Δ is the modeled uncertainty set, structured or unstructured. A second consequence of this fact is that the robust stability margin can be computed by finding $\Delta \in \Delta$ with the smallest $\|\Delta\|_\infty$ such that $\det(I - M_{11}(s_p)\Delta(s_p)) = 0$ for some s_p in the closed RHP. Thus, the **robust stability margin** of generalized LTI uncertainty system, $F_U(M, \Delta)$, is the largest value m such that $F_U(M, \Delta)$ is well-posed and stable for all $\Delta \in \Delta$ with $\|\Delta\|_\infty < m$. Note that if $m > 1$, then $F_U(M, \Delta)$ is robustly stable. Furthermore, one can define the worst-case uncertainty, $\Delta_{w-c} \in \Delta$, which destabilizes $F_U(M, \Delta)$ and has $\|\Delta\|_\infty = m$. Notably, this worst-case uncertainty causes $F_U(M, \Delta)$ to have system poles on the imaginary axis at some $\pm j\omega_{w-c}$.

Robust Stability for Unstructured Uncertainty

The **small-gain theorem (SGT)** provides a method for unstructured complex uncertainty. Namely, if $\Delta(s) \in \mathbb{C}^{n_w \times n_z}$ and $M_{11} \in \mathbb{C}^{n_z \times n_w}$, then $\det(I - M_{11}\Delta) \neq 0$ for all $\Delta \in \mathbb{C}$ with $\bar{\sigma}(\Delta) < m$ if and only if $\bar{\sigma}(M) \leq \frac{1}{m}$. This can be proven by recalling the maximum singular value is equal to the matrix 2,2 induced norm. Thus, for $\bar{\sigma}(M) \leq \frac{1}{m}$, then

$$\bar{\sigma}(M_{11}\Delta) \leq \bar{\sigma}(M)\bar{\sigma}(\Delta) < 1 \quad \forall \Delta \text{ with } \bar{\sigma}(\Delta) < m \quad (3.150)$$

which implies $\|M\Delta\vec{v}\| < \|\vec{v}\|$ for any non-zero $\vec{v} \in \mathbb{C}^{n_v}$ or $(I - M\Delta)\vec{v} \neq 0$, i.e. $\det(I - M\Delta) \neq 0$ for all $\bar{\sigma}(\Delta) < m$ which proves the sufficient condition. Furthermore, if $\bar{\sigma}(M) > \frac{1}{m}$, then the SVD of M provides

$$\bar{\sigma}(M)\vec{v} = M\vec{z} \text{ and } \|\vec{v}\|_2 = \|\vec{z}\|_2 = 1 \quad (3.151)$$

and by selecting $\Delta_0 = \frac{1}{\bar{\sigma}(M)}\vec{z}\vec{v}^* \in \mathbb{C}^{n_z \times n_w}$ which provides $\bar{\sigma}(\Delta_0) = \frac{1}{\bar{\sigma}(M)} < m$, one has $(I - M\Delta_0)\vec{v} = 0$ and $\det(I - M\Delta_0) = 0$ which proves the necessary condition.

The SGT can be extended to unstructured LTI dynamic uncertainty through the fact that complex uncertainty can be interpolated at any finite, non-zero frequency using a stable LTI system and the LTI dynamic uncertainty has norm no larger than the chosen complex uncertainty. Formally, for a frequency, $0 < \omega_0 < \infty$ and a rank-one matrix $\Delta_0 = \vec{z}_0\vec{v}_0^* \in \mathbb{C}^{n_z \times n_v}$ there exists an $n_z \times n_v$ stable, LTI system, $\Delta(s)$ such that $\Delta(j\omega_0) = \Delta_0$ and $\|\Delta\|_\infty \leq \bar{\sigma}(\Delta_0)$. Notably, the equivalence between complex and dynamic uncertainty breaks down at $\omega_0 = 0$ and ∞ as an LTI system with real-valued matrices has a real frequency response at $\omega_0 = 0$ and ∞ , i.e. $\Delta(0), \Delta(\infty) \in \mathbb{R}^{n_z \times n_w}$.

In summary, consider a generalized LTI uncertainty system, $F_U(M, \Delta)$ where Δ is an unstructured, stable LTI system and M is stable. $F_U(M, \Delta)$ is well-posed and stable for all $\|\Delta\|_\infty < m$ if and only if $\|M_{11}\|_\infty \leq \frac{1}{m}$, i.e.

$$\max_{\omega \in \mathbb{R} \cup \{\infty\}} \bar{\sigma}(M_{11}(j\omega)) \leq \frac{1}{m} \quad (3.152)$$

which can be proven using the above results. Notably, if M_{11} achieves its peak gain at $\omega_0 = 0$ or ∞ , then the interpolation is performed at some arbitrarily small or large finite frequency.

Robust Stability for Structured Uncertainty

Notably, a consequence of the SGT is that $\det(I - M_{11}\Delta) \neq 0$ for any $\Delta \in \mathbb{C}^{n_w \times n_z}$ with $\bar{\sigma}(\Delta) < \frac{1}{\bar{\sigma}(M_{11})}$. Thus, one can interpret this fact as an optimization problem, i.e.

$$\begin{aligned} \frac{1}{\bar{\sigma}(M_{11})} &= \min_{\Delta \in \mathbb{C}^{n_w \times n_z}} \bar{\sigma}(\Delta) \\ \text{subject to: } &\det(I - M_{11}\Delta) = 0 \end{aligned} \quad (3.153)$$

Thus, for some structured $\Delta \in \Delta$, one can define the **structured singular value (SSV)** for some $M_{11} \in \mathbb{C}^{n_w \times n_z}$ as

$$\mu_\Delta(M_{11}) = \begin{cases} \left[\min_{\Delta \in \Delta} (\bar{\sigma}(\Delta) \text{ s.t. } \det(I - \Delta M_{11}) = 0) \right]^{-1} \\ 0, \text{ if no } \Delta \in \Delta \text{ causes } \det(I - \Delta M_{11}) = 0 \end{cases} \quad (3.154)$$

where notably the second case *may* occur if Δ consists of only real parametric uncertainties.

With the SSV defined one can state a generalization of the SGT which follows almost directly from this definition. Namely, if $\Delta(s) \in \Delta \subset \mathbb{C}^{n_w \times n_z}$ and $M_{11} \in \mathbb{C}^{n_z \times n_w}$, then $\det(I - M_{11}\Delta) \neq 0$ for all $\Delta \in \Delta$ with $\bar{\sigma}(\Delta) < m$ if and only if $\mu_\Delta(M_{11}) \leq \frac{1}{m}$. Notably, if $\Delta \in \mathbb{C}^{n_w \times n_z}$ is a full complex matrix, then the SGT-based maximum singular value optimization provides the SSV as

$$\mu_\Delta(M_{11}) = \bar{\sigma}(M_{11}) \quad (3.155)$$

However, computing $\mu_\Delta(M_{11})$ for general uncertainty structures is a computationally difficult problem, thus, most implementations of SSV-based robustness bounds use upper and lower bounds on $\mu_\Delta(M_{11})$. Thus, as generally $\Delta \subset \mathbb{C}^{n_w \times n_z}$, a simple upper bound is

$$\mu_\Delta(M_{11}) \leq \bar{\sigma}(M_{11}) \quad (3.156)$$

which can be extended via tighter bounds through D -scalings which account for the non-uniqueness in the LFT representation of M .

A simple lower bound for $\mu_\Delta(M_{11})$ can be computed if $\Delta(s)$ that is a diagonal matrix whose diagonal is a single complex scalar $\delta \in \mathbb{C}$, i.e.

$$\Delta = \{\delta I : \delta \in \mathbb{C}\} \quad (3.157)$$

Substituting this into $I - \Delta M_{11}$, one has

$$I - \Delta M_{11} = I - \delta IM_{11} = \delta \left(\frac{1}{\delta} I - M \right) \quad (3.158)$$

which defines an eigenvalue problem for square M_{11} . Thus, for the simplest structure, one has

$$\mu_\Delta(M_{11}) = \bar{\rho}(M_{11}) = \max_i |\lambda(M_{11})| \quad (3.159)$$

where $\bar{\rho}(M_{11})$ is the **spectral radius** of the matrix, M_{11} , i.e. the largest absolute value of the eigenvalues of M_{11} .

Thus, the SSV can be at least bounded above and below by

$$\bar{\rho}(M_{11}) \leq \mu_\Delta(M_{11}) \leq \bar{\sigma}(M_{11}) \quad (3.160)$$

and in general, numerical methods use versions of power iteration and D -scalings on general M_{11} to numerically compute the SSV bounds as

$$\max_Q \lambda(QM_{11}) \leq \mu_\Delta(M_{11}) \leq \inf_D \bar{\sigma}(DM_{11}D^{-1}) \quad (3.161)$$

Finally, assuming one can use interpolation to replace LTI dynamic uncertainty blocks with corresponding complex parametric uncertainty of the same size, one can state an exact, necessary, and sufficient condition for the robust stability margin of the generalized LTI uncertainty system with structured real and complex parametric uncertainty, and stable, LTI dynamic uncertainty, $\Delta \in \Delta$ as follows. Consider a generalized LTI uncertainty system, $F_U(M, \Delta)$ where Δ is a stable LTI system in the structured set, Δ and M is stable. $F_U(M, \Delta)$ is well-posed and stable for all $\Delta \in \Delta$ with $\|\Delta\|_\infty < m$ if and only if

$$\max_{\omega \in \mathbb{R} \cup \{\infty\}} \mu_\Delta(M_{11}(j\omega)) \leq \frac{1}{m} \quad (3.162)$$

which can be proven using the above results.

3.4 Command Tracking Control System Design

This chapter has provided methods for MIMO LTI feedback control systems to be analyzed in terms of stability, controllability, observability, and robust stability using generalized LTI models. However, the design of control systems varies widely and this section will introduce two additional control systems that augment the LTI or nonlinear plant dynamics to provide an architecture to design optimal MIMO LTI controllers alongside these augmentations, namely, servomechanism augmentation and feedback linearization augmentation. These two augmentations provide the robust servomechanism control system and the dynamic inversion control system, respectively.

Servomechanism Control for Command Tracking

In control system design, one often desires the control system to regulate the tracking error of a reference command, $r(t)$, to zero in the presence of unknown disturbances to the system, $w(t)$. From classical control theory for SISO systems, it is known that integral control action is necessary to achieve this zero steady-state tracking error. By the **internal model principle**, the number of integrators in the open-loop transfer function, i.e. the **system type**, must be greater than or equal to the reference and disturbance signal orders. Thus, if $r(t) = w(t) = 0$, then one needs 0 integrators and one has a type 0 control system, also known as a **regulator**. If $\dot{r}(t) = \dot{w}(t) = 0$, then one needs 1 integrator and one has a type 1 control system. If $\ddot{r}(t) = \ddot{w}(t) = 0$, then one needs 2 integrators and one has a type 2 control system.

For MIMO systems, state feedback controllers act as a type 0 control system, regulate system state to zero, but for any non-zero reference command or disturbance, there will be a steady-state error. Thus, the addition of integral action for state feedback is desirable for tracking constant non-zero reference signals with any potential constant non-zero disturbances. For other types of reference commands and disturbances, one requires additional feedback control considerations for zero-error tracking of an array of reference commands in the presence of an array of disturbances. A standard design approach for achieving this is the robust servomechanism control system which consists of two components, a servomechanism and state feedback. The “robust” terminology comes from its ability to reach zero steady-state tracking error in the presence of specified classes of *disturbances*, a type of uncertainty.

Consider the following continuous-time LTI state-space system

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + B\vec{u} + M\vec{w} \\ \vec{y} &= C\vec{x} + D\vec{u}\end{aligned}\tag{3.163}$$

with an additive unknown bounded disturbance, $\vec{w} \in \mathbb{R}^{n_w}$. In addition, assume that the system is controllable and observable.

Next, assume the reference command is prescribed as some commanded output, $\vec{r}(t) \in \mathbb{R}^{n_y}$, which has the following p^{th} order ODE of the form

$$\vec{r}^{[p]} = \sum_{i=1}^p a_i \vec{r}^{[p-i]}\tag{3.164}$$

where the scalar coefficients, a_i , are known and the superscript $[j]$ denotes the j^{th} derivative. Note that a constant command has the ODE form of $\dot{\vec{r}} = 0$ with $p = 1$ and $a_1 = 0$, a ramp command has the ODE form

of $\ddot{\vec{r}} = 0$ with $p = 2$ and $a_2 = a_1 = 0$, and a sinusoidal command at frequency ω_0 has the ODE form of $\ddot{\vec{r}} = -\omega_0^2 \vec{r}$ with $p = 2$, $a_2 = -\omega_0^2$, and $a_1 = 0$. Likewise, assume that the disturbance inputs have the same p^{th} order ODE form

$$\vec{w}^{[p]} = \sum_{i=1}^p a_i \vec{w}^{[p-i]} \quad (3.165)$$

with unknown $w(0) = w_0$.

Also, define the tracking error signal as

$$\vec{e} = \vec{y} - \vec{r} \quad (3.166)$$

which can be differentiated p times to obtain the error ODE

$$\vec{e}^{[p]} - \sum_{i=1}^p a_i \vec{e}^{[p-i]} = \left(\vec{y}^{[p]} - \sum_{i=1}^p a_i \vec{y}^{[p-i]} \right) - \left(\vec{r}^{[p]} - \sum_{i=1}^p a_i \vec{r}^{[p-i]} \right) \quad (3.167)$$

where, by definition, the second term on the right side will be zero and the first term can be written using the output equation and its derivatives as

$$\vec{e}^{[p]} - \sum_{i=1}^p a_i \vec{e}^{[p-i]} = C \left(\vec{x}^{[p]} - \sum_{i=1}^p a_i \vec{x}^{[p-i]} \right) + D \left(\vec{u}^{[p]} - \sum_{i=1}^p a_i \vec{u}^{[p-i]} \right) \quad (3.168)$$

which represents a set of coupled ODEs for the error differential equation.

Then, defining

$$\vec{\eta} = \vec{x}^{[p]} - \sum_{i=1}^p a_i \vec{x}^{[p-i]} \quad (3.169)$$

and

$$\vec{\mu} = \vec{u}^{[p]} - \sum_{i=1}^p a_i \vec{u}^{[p-i]} \quad (3.170)$$

the error differential equation becomes

$$\vec{e}^{[p]} - \sum_{i=1}^p a_i \vec{e}^{[p-i]} = C \vec{\eta} + D \vec{\mu} \quad (3.171)$$

Differentiating, one has

$$\dot{\vec{\eta}} = \vec{x}^{[p+1]} - \sum_{i=1}^p a_i \vec{x}^{[p-i+1]} \quad (3.172)$$

and by substitution for $\dot{\vec{x}}$, one has

$$\dot{\vec{\eta}} = A \left(\vec{x}^{[p]} - \sum_{i=1}^p a_i \vec{x}^{[p-i]} \right) + B \left(\vec{u}^{[p]} - \sum_{i=1}^p a_i \vec{u}^{[p-i]} \right) + M \left(\vec{w}^{[p]} - \sum_{i=1}^p a_i \vec{w}^{[p-i]} \right) \quad (3.173)$$

or

$$\dot{\vec{\eta}} = A \vec{\eta} + B \vec{\mu} \quad (3.174)$$

which is the original system without the additive disturbance term.

Finally, the **servomechanism state-space model**

$$\dot{\vec{z}} = \tilde{A} \vec{z} + \tilde{B} \vec{\mu} \quad (3.175)$$

where \vec{z} is the augmented $n_x + p \times n_y$ state vector

$$\vec{z} = \begin{bmatrix} \vec{e} \\ \vec{e} \\ \vdots \\ \vec{e}^{[p-1]} \\ \vec{\eta} \end{bmatrix} \quad (3.176)$$

with augmented state matrix

$$\tilde{A} = \begin{bmatrix} 0 & I & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & I & 0 \\ a_p I & a_{p-1} I & \cdots & a_1 I & C \\ 0 & 0 & \cdots & 0 & A \end{bmatrix} \quad (3.177)$$

and augmented input matrix

$$\tilde{B} = \begin{bmatrix} 0 \\ \vdots \\ 0 \\ D \\ B \end{bmatrix} \quad (3.178)$$

In steady-state, this allows the state vector, \vec{x} , to be nonzero in which case

$$\vec{r} = C \vec{x} + D \vec{u} \quad (3.179)$$

Lastly, note that to check for controllability of the servomechanism state-space model, one requires that (A, B) be controllable, $n_u \geq n_y$, and that (A, B, C, D) must not have any zeros in common with the ODE for \vec{y} .

For this system, one can design a servomechanism state feedback control law as

$$\vec{\mu} = K_z \vec{z} \quad (3.180)$$

where K_z is the feedback gain matrix which can be partitioned in the same manner as \vec{z} , i.e.

$$K_z = [K_p \ K_{p-1} \ \cdots \ K_1 \ K_x] \quad (3.181)$$

Next, recalling the equation for $\vec{\mu}$, one has

$$\vec{\mu} = \vec{u}^{[p]} - \sum_{i=1}^p a_i \vec{u}^{[p-i]} = [K_p \ K_{p-1} \ \cdots \ K_1 \ K_x] \begin{bmatrix} \vec{e} \\ \vec{e} \\ \vdots \\ \vec{e}^{[p-1]} \\ \vec{\eta} \end{bmatrix} \quad (3.182)$$

or, multiplying out and substituting for $\vec{\eta}$, one has

$$\vec{u} = \vec{u}^{[p]} - \sum_{i=1}^p a_i \vec{u}^{[p-i]} = \sum_{i=1}^p K_i \vec{e}^{[p-i]} + K_x (\vec{x}^{[p]} - \sum_{i=1}^p a_i \vec{x}^{[p-i]}) \quad (3.183)$$

Then, integrating p times provides the control law for the original system as

$$\vec{u} = K_x \vec{x} + \sum_{i=1}^p \frac{(a_i(\vec{u} + K_x \vec{x}) + K_i \vec{e})}{s^i} \quad (3.184)$$

where it should be noted that the K_x term will enforce closed-loop stability to the dynamical system and the p integrators and the integrator gains provide integral error control, and the coefficients, a_i , embed the model of the reference command to be tracked. Thus, the closed-loop dynamics model is

$$\dot{\vec{x}}_{aug} = (\tilde{A} + \tilde{B}K_z) \vec{x}_{aug} + \begin{bmatrix} -I_{n_y \times n_y} \\ 0_{n_x \times n_y} \end{bmatrix} \vec{r} \quad (3.185)$$

where \vec{x}_{aug} is the **augmented state** which is related to the servomechanism state, \vec{z} , by p integrations, i.e.

$$\vec{x}_{aug} = \frac{\vec{z}}{s^p} \quad (3.186)$$

Model-Following Control for Command Tracking

3.5 LTV Systems Theory and Control

A **time-varying dynamical system** is given generally by

$$\begin{aligned} \dot{\vec{x}}(t) &= f(t, \vec{x}, \vec{u}) \\ \vec{y}(t) &= h(t, \vec{x}, \vec{u}) \end{aligned} \quad (3.187)$$

which can also be unforced, i.e.

$$\begin{aligned} \dot{\vec{x}}(t) &= f(t, \vec{x}) \\ \vec{y}(t) &= h(t, \vec{x}) \end{aligned} \quad (3.188)$$

Suppose that one initializes at time $t_0 \geq 0$ the state of a dynamical system as some $\vec{x}(t_0) = \vec{x}_0 \in \mathbb{R}^n$. Then, one has an **initial value problem (IVP)**, also known as the **Cauchy problem**, which may have many solutions, one unique solution, or no existing solution. Contrary to LTI dynamical systems, the existence and uniqueness of solutions to IVPs for non-LTI dynamical systems is not always guaranteed and often one can only provide sufficient conditions for existence and uniqueness.

The **Cauchy-Peano Theorem** states that sufficient conditions for the IVP to admit a solution (not may be unique) if, for some $T > 0$, some $\epsilon > 0$, and $f(t, \vec{x}) : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is continuous in a closed region, i.e.

$$B = \{(t, \vec{x}) : |t - t_0| \leq T, \|\vec{x} - \vec{x}_0\| \leq \epsilon\} \subseteq \mathbb{R} \times \mathbb{R}^n \quad (3.189)$$

then there exists $t_0 < t_1 \leq T$ such that the IVP has at least one continuously differentiable solution $\vec{x}(t)$ on the interval $[t_0, T]$. The assumed continuity of $f(t, \vec{x})$ in its arguments, t and \vec{x} ensures that there is at least one solution of the IVP. Note that this theorem does not guarantee the uniqueness of the solution.

The key constraint that yields uniqueness is the **Lipschitz condition** whereby $f(t, \vec{x})$ satisfies the inequality

$$\|f(t, \vec{x}) - f(t, \vec{y})\| \leq L\|\vec{x} - \vec{y}\| \quad (3.190)$$

for all (t, \vec{x}) and (t, \vec{y}) in some “neighborhood” of (t_0, \vec{x}_0) with a finite constant $L > 0$. A theorem for sufficient conditions for the IVP to admit the *local* existence and uniqueness of a solution states that if, for some $\epsilon > 0$, $f(t, \vec{x}) : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is piece-wise continuous in t and satisfies the Lipschitz condition, i.e.

$$\forall \vec{x}, \vec{y} \in B = \{\vec{x} \in \mathbb{R}^n : \|\vec{x} - \vec{x}_0\| \leq \epsilon\}, \forall t \in [t_0, t_1] \quad (3.191)$$

then, there exists some $\delta > 0$ such that the IVP for the state equation $\dot{\vec{x}} = f(t, \vec{x})$ with $\vec{x}(t_0) = \vec{x}_0$ has a unique solution over $[t_0, t_0 + \delta]$. A theorem for sufficient conditions for the IVP to admit the *global* existence and uniqueness of a solution states that if, for some finite $L > 0$ and $f(t, \vec{x}) : \mathbb{R} \times \mathbb{R}^n \rightarrow \mathbb{R}^n$ is piece-wise continuous in t and is globally Lipschitz in \vec{x} , i.e.

$$\|f(t, \vec{x}) - f(t, \vec{y})\| \leq L\|\vec{x} - \vec{y}\|, \forall \vec{x}, \vec{y} \in \mathbb{R}^n, \forall t \in [t_0, t_1] \quad (3.192)$$

then, the IVP has a unique solution over $[t_0, t_1]$ where the final time t_1 may be arbitrarily large.

Equilibrium Points for Time-Varying Systems

For unforced, time-varying dynamics functions $f : [0, \infty) \times D \rightarrow \mathbb{R}^n$ as piece-wise continuous in t and locally Lipschitz in \vec{x} and with domain $D \subset \mathbb{R}^n$ that contains the origin $\vec{x} = \vec{0}$. Then, the origin in \mathbb{R}^n is an **equilibrium point** for the unforced, time-varying system at $t_0 = 0$ if

$$f(t, \vec{0}) = \vec{0}, \forall t \geq 0 \quad (3.193)$$

Without loss of generality, this result can be extended beyond the origin and initial time by defining a nonzero vector $\vec{x} \in \mathbb{R}^n$ to be an equilibrium point of $\dot{\vec{x}} = f(t, \vec{x})$ at a nonzero initial time $t = t_0$, thus

$$f(t, \vec{x}) = \vec{0}, \forall t \geq t_0 \quad (3.194)$$

then defining the new time as $\tau = t - t_0$ and new state as $\vec{z}(\tau) = \vec{x}(\tau + t_0) - \vec{x}$, one has for the new system dynamics

$$\frac{d\vec{z}(\tau)}{d\tau} = \frac{\vec{x}(\tau + t_0)}{dt} = f(\tau + t_0, \vec{z}(\tau) + \vec{x}) = g(\tau, \vec{z}(\tau)) \quad (3.195)$$

with $g(0, \vec{0}) = f(t_0, \vec{x}) = 0$. Thus, one can shift the equilibrium point to the origin and initial time to zero.

Furthermore, suppose one has a state trajectory $\vec{x}(t)$ that starts at $t = t_0$, i.e.

$$\dot{\vec{x}}^*(t) = f(t, \vec{x}(t)), \forall t \geq t_0 \quad (3.196)$$

then again defining the new time as $\tau = t - t_0$ and the new state this time as $\vec{z}(\tau) = \vec{x}(\tau + t_0) - \vec{x}(\tau + t_0)$, one has for the new system dynamics

$$\begin{aligned} \frac{d\vec{z}(\tau)}{d\tau} &= \frac{\vec{x}(\tau + t_0)}{dt} - \frac{d\vec{x}(\tau + t_0)}{dt} \\ &= f(\tau + t_0, \vec{z}(\tau) + \vec{x}(\tau + t_0)) - f(\tau + t_0, \vec{z}(\tau) + \vec{x}(\tau + t_0)) = g(\tau, \vec{z}(\tau)) \end{aligned} \quad (3.197)$$

with $g(0, \vec{0}) = \vec{0}$. Consequently, these new dynamics around the origin as an equilibrium point while starting at t_0 allows one to determine the original system behavior around the original nonzero equilibrium \vec{x} , i.e. one can assess the system relative dynamics with respect to any time-dependent trajectory $\vec{x}(t)$, starting at an arbitrary initial time $t_0 \geq 0$. This is key for stability analysis of equilibrium points.

Lyapunov Stability Theory for Time-Varying Systems

Lyapunov stability theory provides an analysis tool for assessing the behavior of system trajectories near equilibrium points, but *without* explicit computation of these trajectories. System stability can be interpreted as a continuity of the system trajectories, with respect to initial conditions, over an *infinite* time interval. This infinite time interval highlights the primary notion of stability as a continuity property of Lipschitz-continuous differential equations holding infinitely in time. Formally, this can be stated by letting $\vec{x}(t, \vec{x}_0)$ define a unique solution of $\dot{\vec{x}} = f(t, \vec{x})$ with initial condition $\vec{x}(t_0) = \vec{x}_0$ which exists on a finite, possibly open-ended interval $[t_0, T]$. The continuity property of $\vec{x}(t, \vec{x}_0)$ due to changes in \vec{x}_0 can be stated as follows. Given any constant $\epsilon > 0$, there must exist a sufficiently small constant $\delta > 0$ such that for all perturbed initial conditions $\vec{x}_0 + \Delta \vec{x}_0$ with $\|\Delta \vec{x}_0\| \leq \delta$, the corresponding perturbed solution $\vec{x}(t, \vec{x}_0 + \Delta \vec{x}_0)$ deviates from the original \vec{x}_0 by no more than ϵ , i.e. $\|\vec{x}(t, \vec{x}_0 + \Delta \vec{x}_0) - \vec{x}(t, \vec{x}_0)\| \leq \epsilon$, for all $t_0 \leq t < T$.

The **Lyapunov stability of an equilibrium point**, $\vec{x} = \vec{0}$, i.e. the origin, for time-varying unforced dynamics as *stable* if for any $\epsilon > 0$ and $t \geq 0$ there exists some $\delta(\epsilon, t_0) > 0$ such that for all initial conditions $\|\vec{x}_0\| < \delta$ and for all $t \geq t_0 \geq 0$, the corresponding system trajectories are bounded, i.e. $\|\vec{x}(t)\| < \epsilon$, otherwise it is *unstable*. In essence, Lyapunov stability of an equilibrium point \vec{x} means that given an outer “hyper-sphere” $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$, one can find an inner “hyper-sphere” $B_\delta = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \delta\}$, such that any trajectory that starts inside B_δ will evolve inside B_ϵ for *all* future times. Nonlinear dynamical systems may display completely different behavior in various domains of the state. Thus, one must distinguish between local and global stability. In particular, the equilibrium point, $\vec{x} = \vec{0}$, i.e. the origin, has **global stability** if it is stable and $\lim_{\epsilon \rightarrow \infty} \delta(\epsilon, t_0) = \infty$, i.e. the trajectory will not deviate “too far” from any arbitrary initial conditions. System trajectories of time-varying dynamical systems depend on initial time t_0 and the stability of an equilibrium point for time-varying systems may depend on t_0 . Thus, the equilibrium point has **uniform stability** if it is stable and δ does not depend on t_0 .

Beyond these, one can also define the following stricter senses of stability. The equilibrium point, $\vec{x} = \vec{0}$, i.e. the origin, has **asymptotic stability** if it is stable and there exists a constant $c = c(t_0) > 0$ such that $\vec{x} \rightarrow \vec{0}$ as $t \rightarrow \infty$ for all $\|\vec{x}\| \leq c$. The equilibrium point, $\vec{x} = \vec{0}$, i.e. the origin, has **uniform asymptotic stability** if it is uniformly stable and there exists a constant $c > 0$ independent of t_0 such that $\vec{x} \rightarrow \vec{0}$ as $t \rightarrow \infty$ for all $\|\vec{x}\| \leq c$ uniformly in t_0 . The equilibrium point, $\vec{x} = \vec{0}$, i.e. the origin, has **global uniform asymptotic stability** if it is uniformly asymptotically stable and $\lim_{\epsilon \rightarrow \infty} \delta(\epsilon) = \infty$. Of note, uniform asymptotic stability is typically a highly desirable property of control system design as these systems are able to maintain their closed-loop performance in the presence of state perturbations and disturbances.

In his dissertation on the stability of motion, Lyapunov developed two theorems known as the indirect and direct methods for assessing the stability of nominal solutions to dynamical systems governed by a finite number of coupled ODEs. Both methods provide verifiable sufficient conditions for stability of a nominal trajectory without an explicit knowledge of the system solutions. **Lyapunov's indirect method** states that one can determine the stability of an equilibrium point, i.e. the origin, for a nonlinear, time-invariant n -dimensional systems by linearizing the system dynamics about the equilibrium point. This method has

been utilized in previous systems theory discussion in this textbook. Notably, **Lyapunov's direct method** requires the concepts of positive and negative definite functions and the time derivative of a scalar function along the state trajectories of a differential equation, i.e. it's possible solutions.

A scalar function $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ of a vector argument $\vec{x} \in \mathbb{R}^n$ is called **locally positive definite** if $V(\vec{0}) = 0$ and there exists a constant ϵ such that $V > 0$ for all $\vec{x} \in \mathbb{R}^n$ in the neighborhood of the origin, i.e. $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$, where if $\epsilon = \infty$, then $V(\vec{x})$ is **globally positive definite**.

A scalar function $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ of a vector argument $\vec{x} \in \mathbb{R}^n$ is called **locally positive semi-definite** if $V(\vec{0}) = 0$ and there exists a constant ϵ such that $V \geq 0$ for all $\vec{x} \in \mathbb{R}^n$ in the neighborhood of the origin, i.e. $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$, where if $\epsilon = \infty$, then $V(\vec{x})$ is **globally positive semi-definite**.

A scalar function $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ of a vector argument $\vec{x} \in \mathbb{R}^n$ is called **locally negative definite** if $V(\vec{0}) = 0$ and there exists a constant ϵ such that $V < 0$ for all $\vec{x} \in \mathbb{R}^n$ in the neighborhood of the origin, i.e. $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$, where if $\epsilon = \infty$, then $V(\vec{x})$ is **globally negative definite**.

A scalar function $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ of a vector argument $\vec{x} \in \mathbb{R}^n$ is called **locally negative semi-definite** if $V(\vec{0}) = 0$ and there exists a constant ϵ such that $V \leq 0$ for all $\vec{x} \in \mathbb{R}^n$ in the neighborhood of the origin, i.e. $B_\epsilon = \{\vec{x} \in \mathbb{R}^n : \|\vec{x}\| \leq \epsilon\}$, where if $\epsilon = \infty$, then $V(\vec{x})$ is **globally negative semi-definite**.

Given a scalar continuously differentiable function $V(\vec{x})$ where $\vec{x}(t) \in \mathbb{R}^n$ represents a time-varying trajectory of the time-varying system. Then, the time derivative of $V(\vec{x}(t))$ along the system solution $\vec{x}(t) = [x_1(t), \dots, x_n(t)]^T$ as

$$\dot{V}(\vec{x}) = \sum_{i=1}^n \frac{\partial V}{\partial x_i} \dot{x}_i = \sum_{i=1}^n \frac{\partial V}{\partial x_i} f_i(t, \vec{x}) = \nabla V(\vec{x}) f(t, \vec{x}) \quad (3.198)$$

where $\nabla V(\vec{x}) = [\frac{\partial V}{\partial x_1}, \dots, \frac{\partial V}{\partial x_n}]$ is the row vector gradient of $V(\vec{x})$ with respect to \vec{x} . Note that the time derivative of $V(\vec{x})$ depends not only on the function $V(\vec{x})$ but also on the system dynamics under consideration. Changing the latter while keeping the same $V(\vec{x})$ may result in a different $\dot{V}(\vec{x})$. Next, to state Lyapunov's direct method, let $\vec{x} = \vec{0} \in \mathbb{R}^n$ be an equilibrium point for the time-varying dynamics, whose initial conditions are drawn from a domain $D \subset \mathbb{R}^n$ with $\vec{x} \in D$ and $t_0 = 0$.

Then, if on the domain D , there exists a continuously differentiable locally positive definite function $V(\vec{x}) : D \rightarrow \mathbb{R}$, whose time derivative along the system trajectories is locally negative semi-definite, i.e.

$$\dot{V}(\vec{x}) = \nabla V(\vec{x}) f(t, \vec{x}) \leq 0 \quad (3.199)$$

for all $t \geq 0$ and for all $\vec{x} \in D$, then $\vec{x} = \vec{0}$ is locally uniformly stable. Furthermore, if $\dot{V}(\vec{x}) < 0$ for all $t \geq 0$, i.e. the time derivative along the system trajectories is locally negative definite, then $\vec{x} = \vec{0}$ is locally uniformly asymptotically stable. Here any locally positive definite $V(\vec{x})$ is called a **Lyapunov function candidate** and if it satisfies the time derivative condition it is called a **Lyapunov function**. Note though that the existence of a Lyapunov function is sufficient to claim uniform stability for the equilibrium point, if one cannot be found, nothing can be stated about the stability of the equilibrium point. Furthermore, it should be noted that Lyapunov functions are not unique.

The Lyapunov function can be viewed as an “energy-like” function for testing the stability of a system. If the values of V do not increase along the system trajectories, then the origin is uniformly stable. If V strictly decreases, then, in addition, the system trajectories will approach the origin asymptotically. Lastly, note that the uniform asymptotic stability requires a subset of D known as the **region of attraction**, i.e. starting there the system solutions will converge to the origin. If the region of attraction of a uniformly asymptotically stable equilibrium is \mathbb{R}^n , then the equilibrium is said to be globally uniformly asymptotically stable.

Next, define the property $\lim_{\|\vec{x}\| \rightarrow \infty} V(\vec{x}) = \infty$ where $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ as a **radially unbounded** Lyapunov function candidate. In addition, define $V_c = \{\vec{x} \in \mathbb{R}^n : V(\vec{x}) = c\}$ as a **level set**, i.e. V_c has a constant value c , of a radially unbounded Lyapunov function candidate $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$, and define $\Omega_c = \{\vec{x} \in \mathbb{R}^n : V(\vec{x}) \leq c\}$ as the union of the interior set of V_c and V_c itself.

Then, consider a converging sequence $\lim_{k \rightarrow \infty} \vec{x}_k = \vec{a}$ with all \vec{x} from Ω_c , then the limit point \vec{a} must also be in Ω_c . Here, Ω_c is a **closed set** because $V(\vec{x})$ is continuous on \mathbb{R}^n and $V(\vec{x}) \leq c$ for all $k = 1, 2, \dots$, thus $c \geq \lim_{k \rightarrow \infty} V(\vec{x}_k) = V(\vec{a})$ and $\vec{a} \in \Omega_c$ which shows every converging sequence in Ω_c has its limit point in the same set, i.e. closed. Ω_c is also a **bounded set**. If it was not, then there must exist a sequence of points $\{\vec{x}_k\} \in \Omega_c$ whose limit is ∞ . However, since $V(\vec{x})$ is continuous and radially unbounded, then $c \geq \lim_{k \rightarrow \infty} V(\vec{x}_k) = \infty$, which is a contradiction. Thus, since Ω_c is closed, bounded, and belongs to \mathbb{R}^n , it is a **compact set** which allows one to state the following **Krasovskii-LaSalle theorem**. If $\vec{x} = \vec{0}$ is an equilibrium point of $\dot{\vec{x}} = f(t, \vec{x})$ and $V(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}$ is a radially unbounded Lyapunov function, then \vec{x} is a globally uniformly asymptotically stable equilibrium point. Note that a simple example of radially unbounded Lyapunov function candidates include the quadratic form $V(\vec{x}) = \vec{x}^T P \vec{x}$ where P is a symmetric positive definite matrix, i.e. $P = P^T > 0$.

Linear, Time-Varying Systems

A linear, time-varying (LTV) systems are represented by the following linear state-space model

$$\begin{aligned}\dot{\vec{x}}(t) &= A(t)\vec{x}(t) + B(t)\vec{u}(t) \\ \vec{y}(t) &= C(t)\vec{x}(t) + D(t)\vec{u}(t)\end{aligned}\tag{3.200}$$

Similar to LTI systems, the general solution of a LTV system can be written as

$$\vec{x}(t) = \Phi(t, t_0)\vec{x}(t_0) + \int_{t_0}^t \Phi(t, \tau)B(\tau)\vec{u}(\tau)d\tau\tag{3.201}$$

where $\Phi(t, t_0)$ is the **state-transition matrix** of LTV system and can be given by the **Peano-Baker series** form

$$\Phi(t, t_0) = I + \int_{\tau}^t A(\sigma_1)d\sigma_1 + \int_{\tau}^t A(\sigma_1) \int_{\tau}^{\sigma_1} A(\sigma_2)d\sigma_2 d\sigma_1 + \dots\tag{3.202}$$

which converges uniformly and absolutely to a solution that exists and is unique. The $\vec{\beta}(t)$ typically also has restricted admissible trajectories for any particular realization.

For Lyapunov stability of LPV systems, one can use the Lyapunov candidate function with $P > 0$

$$V(\vec{x}) = \vec{x}^T P \vec{x}\tag{3.203}$$

which has time derivative

$$\dot{V}(\vec{x}) = \dot{\vec{x}}^T P \vec{x} + \vec{x}^T P \dot{\vec{x}}\tag{3.204}$$

and evaluated along the state trajectories, one has

$$\dot{V}(\vec{x}) = \vec{x}^T (A^T(t)P + PA(t))\vec{x}\tag{3.205}$$

which is negative definite, if and only if

$$A^T(t)P + PA(t) < 0 \quad (3.206)$$

for all t .

Define the **linear, time-varying controllability Gramian** using the state-transition matrix as

$$W_C(t_0, t_1) = \int_{t_0}^{t_1} \Phi(t_0, t)B(t)B(t)^T\Phi(t_0, t)^T dt \quad (3.207)$$

where $W_C(t_0, t_1)$ is symmetric, positive semi-definite, and satisfies

$$W_C(t_0, t_1) = W_C(t_0, t) + \Phi(t_0, t)W(t, t_1)\Phi(t_0, t)^T \quad (3.208)$$

If there exists some t_1 such that $W_C(t_0, t_1)$ is full rank, then the LTV system is state controllable.

Define the **linear, time-varying observability Gramian** using the state-transition matrix as

$$W_O(t_0, t_1) = \int_{t_0}^{t_1} \Phi(t, t_0)^TC(t)^TC(t)\Phi(t, t_0)dt \quad (3.209)$$

where $W_O(t_0, t_1)$ is symmetric, positive semi-definite, and satisfies

$$W_O(t_0, t_1) = W_O(t_0, t) + \Phi(t, t_0)^TW(t, t_1)\Phi(t, t_0) \quad (3.210)$$

If there exists some t_1 such that $W_O(t_0, t_1)$ is full rank, then the LTV system is observable.

A special class of LTV systems are **linear, parameter-varying (LPV)** systems, represented by the following linear state-space model

$$\begin{aligned} \dot{\vec{x}}(t) &= A(\vec{\beta}(t))\vec{x}(t) + B(\vec{\beta}(t))\vec{u}(t) \\ \vec{y}(t) &= C(\vec{\beta}(t))\vec{x}(t) + D(\vec{\beta}(t))\vec{u}(t) \end{aligned} \quad (3.211)$$

where $\vec{\beta}(t)$ is the **parameter vector** that must be a continuously differentiable function of time and the state-space matrices are also continuous functions of $\vec{\beta}$. LPV system models typically arise when one linearizes a nonlinear system about equilibrium points specified by certain parameters, i.e. $\vec{\beta}$.

As mentioned previously, a common control design approach for LPV systems is known as **gain scheduling**. Here, one computes an array of optimal LTI controllers at a variety of *different* equilibrium points across the ranges of admissible $\vec{\beta}$, e.g. obtain *different* optimal LTI feedback gain matrices. Then, one can blend the various LTI controllers to obtain a single, integrated control design, e.g. interpolate between the different optimal LTI feedback gain matrices if one is operating away from the equilibrium points. Gain scheduling works well for dynamical systems that are locally “smooth,” i.e. the linearized dynamics are reasonably accurate near a given equilibrium condition, but the nonlinearities have significant effect over the entire range of $\vec{\beta}$.

Continuous-Time Linear Optimal Control Theory

4.1 Introduction to Optimal Control

Decision theory is the determination of the optimal decision given constraints and assumptions on a decision problem. **Optimal control theory** is special case of a model-based decision problem and can be stated as deciding which controller one should select as the optimal control policy with respect to some chosen objective for the dynamical system. When the chosen objective involves the actions of other, separate decision maker(s), one has an extension of optimal control theory called **differential game theory**. Regardless, the dynamic optimization, i.e. optimization over time, is critical to optimal control theory and can be formulated for both continuous-time and discrete-time dynamical systems. This chapter focuses on the continuous-time framework and a similar framework for discrete-time dynamical systems can be found in Appendix B of this textbook.

To solve for this optimal control policy, one uses methods from **mathematical optimization**, also known as **mathematical programming**, which formulates the selection of the “best” or optimal element of a set of possible elements with respect to some criteria, i.e. objective. Typically, the optimal is defined as the decision that achieves the minimum (or maximum) with respect to the mathematical representation of the objective. These sets of possible elements can be continuous, discrete, and/or constrained. The objective in optimal control can be imposed superficially by the control designer or dictated by the real dynamic process. Furthermore, this optimization procedure provides the general **optimal control problem (OCP)** which is typically stated as a minimization as follows.

The OCP for continuous-time dynamical systems can be generally written as

$$\begin{aligned} \vec{u}_{opt}(t) = & \underset{u(t) \forall t \in [t_0, t_f]}{\operatorname{argmin}} \mathcal{J}(\vec{x}, \vec{u}, t, t_0, t_f) \\ & \text{subject to:} \\ & \text{continuous-time dynamics: } \dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \\ & \text{boundary conditions: } e(\vec{x}(t_0), t_0, \vec{x}(t_f), t_f) = 0 \\ & \text{path constraints: } c(\vec{x}(t), \vec{u}(t), t) \leq 0 \end{aligned} \tag{4.1}$$

where $\vec{u}^{\text{opt}}(t)$ is the **optimal control function**, argmin stands for “the argument which minimizes” the following expression, t_0 is the start time, t_f is the final time, $t_f - t_0$ is the **time horizon**, $\mathcal{J}(\vec{x}, \vec{u}, t)$ is the **cost functional**, also known as the **objective functional** or the **performance index**, which generally depends on the state, input, and time. The term **functional** is a mathematical definition for a “function of a function.”

In optimal control design, one determines what to use for the cost functional, \mathcal{J} . Next, one must show there exists an optimal solution. Then, one uses synthesis methods to solve for the optimal controller based on the chosen \mathcal{J} . The optimization is also subject to some dynamics for the state of the system, boundary conditions for the system, and some constraints on the state, inputs, and time and there may be multiple solutions to the OCP. Lastly, it should be noted often one is not necessarily interested in finding the truly optimal solution to the OCP, but an efficient near-optimal solution for complex OCPs.

Within the general OCP framework, there are different versions of the OCP that can be further characterized. First, one can optimize for finite- or infinite-time horizons typically shortened to finite- or infinite-horizon OCPs, i.e. $t_f \neq \infty$ or $t_f = \infty$, respectively. Also, often the infinite-horizon OCPs may be easier to solve than finite-horizon OCPs as the optimal control policy will not depend on any specific time, but only on the state, thus becoming a *fixed-gain* control policy, which may be desirable. It should also be mentioned that finite-horizon OCPs may be re-optimized recursively by the control system forming a **receding-horizon control (RHC)**, also known as a **model predictive control (MPC)**. Second, the dynamics of the system may be linear or nonlinear where linear systems allow for simpler solvers. Third, OCP solvers may be simplified if the OCP is unconstrained or constrained. The effects of constraints on the solving method typically revolves around whether there are state constraints, input constraints, or both. Finally, the cost functional or function may be characterized as linear, quadratic, convex, or non-convex, each requiring more complex solvers. This is due to the fact that for a finite minimum cost to exist, the simplest model would be a linear cost for constrained OCPs and a quadratic cost for unconstrained OCPs. Beyond these cases, one can also make a more general distinction between convex and non-convex costs since convexity implies a local minimum is a global minimum which simplifies many numerical search methods which modern computing can perform very efficiently.

Generalized Calculus of Variations

Consider the **classical OCP**

$$\vec{u}_{opt}(t) = \underset{u(t) \forall t \in [t_0, t_f]}{\operatorname{argmin}} \mathcal{E}(\vec{x}(t_f)) + \int_{t_0}^{t_f} \mathcal{L}(\vec{x}(t), \vec{u}(t)) dt$$

subject to:

$$\text{continuous-time dynamics: } \dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t))$$

$$\text{initial conditions: } \vec{x}(t_0) - \vec{x}_0 = 0$$

where t_f and t_0 are fixed, \mathcal{E} is the **Mayer term**, also known as the **endpoint cost** or **terminal cost**, and \mathcal{L} is the **Lagrangian term**, also known as the **running cost** or the **instant cost** which must be nonnegative.

To solve these OCPs, one can use **generalized calculus of variations** by augmenting the Lagrangian with the **costate vector**, also known as the **adjoint vector**, $\vec{\lambda}(t)$, which are similar to Lagrange multiplier, but are functions of time rather than constants. These adjoint vectors allow one to form the augmented cost functional

$$\bar{\mathcal{J}} = \mathcal{E}(\vec{x}(t_f)) + \int_{t_0}^{t_f} \left(\mathcal{L}(\vec{x}(t), \vec{u}(t), t) + \vec{\lambda}^T(f(\vec{x}(t), \vec{u}(t), t) - \dot{\vec{x}}(t)) dt \right) \quad (4.3)$$

where $\vec{\lambda}(t)$ can be *any* vector because the state dynamics require that

$$f(\vec{x}(t), \vec{u}(t)) - \dot{\vec{x}}(t) = 0 \quad (4.4)$$

holds for all time, which means $\vec{\lambda}(t)$ is being multiplied by zero in the expression above. Furthermore, define the **Hamiltonian** as

$$\mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t), t) = \mathcal{L}(\vec{x}(t), \vec{u}(t)) + \vec{\lambda}^T f(\vec{x}(t), \vec{u}(t)) \quad (4.5)$$

where notably

$$\dot{\vec{x}}^T = f(\vec{x}(t), \vec{u}(t))^T = \frac{\partial \mathcal{H}}{\partial \vec{\lambda}} \quad (4.6)$$

Thus, one has

$$\bar{\mathcal{J}} = \mathcal{E}(\vec{x}(t_f)) + \int_{t_0}^{t_f} \mathcal{H} - \vec{\lambda}^T \dot{\vec{x}}(t) dt \quad (4.7)$$

Next, one can form the **variation of $\bar{\mathcal{J}}$** as

$$\delta \bar{\mathcal{J}} = \frac{\partial \mathcal{E}}{\partial \vec{x}} \partial \vec{x}(t_f) + \int_{t_0}^{t_f} \frac{\partial \mathcal{H}}{\partial \vec{x}} \partial \vec{x} + \frac{\partial \mathcal{H}}{\partial \vec{u}} \partial \vec{u} - \vec{\lambda}^T \partial \dot{\vec{x}} dt \quad (4.8)$$

By expanding the last term using integration by parts, one obtains

$$-\int_{t_0}^{t_f} \vec{\lambda}^T \partial \dot{\vec{x}} dt = -\vec{\lambda}^T(t_f) \partial \vec{x}(t_f) + \vec{\lambda}^T(t_0) \partial \vec{x}(t_0) + \int_{t_0}^{t_f} \vec{\lambda}^T \partial \vec{x} dt \quad (4.9)$$

By substitution and rearrangement, one can separate $\delta \bar{\mathcal{J}}$ into four different components as

$$\delta \bar{\mathcal{J}} = \left(\frac{\partial \mathcal{E}(t_f)}{\partial \vec{x}} - \vec{\lambda}^T(t_f) \right) \partial \vec{x}(t_f) + \vec{\lambda}^T(t_0) \partial \vec{x}(t_0) + \int_0^{t_f} \left(\frac{\partial \mathcal{H}}{\partial \vec{x}} + \vec{\lambda}^T \right) \partial \vec{x} + \frac{\partial \mathcal{H}}{\partial \vec{u}} \partial \vec{u} dt \quad (4.10)$$

However, as the adjoint vectors are arbitrary, one can select them to make the coefficients of $\partial \vec{x}(t)$ and $\partial \vec{x}(t_f)$ equal to zero, i.e.

$$\begin{aligned}\dot{\vec{\lambda}}^T &= -\frac{\partial \mathcal{H}}{\partial \vec{x}} \\ \vec{\lambda}(t_f) &= \frac{\partial \mathcal{E}(t_f)}{\partial \vec{x}}\end{aligned}\quad (4.11)$$

where the first equation is the **costate equation**, also known as the **adjoint equation**, for the dynamics of $\vec{\lambda}$ while the second provides the final condition for $\vec{\lambda}$.

With this choice, one has

$$\delta \bar{\mathcal{J}} = \int_0^{t_f} \frac{\partial \mathcal{H}}{\partial \vec{u}} \delta \vec{u} dt \quad (4.12)$$

which for \mathcal{J} to be minimized requires $\delta \bar{\mathcal{J}} = 0$, thus, assuming \vec{u} is free to vary, one requires

$$\frac{\partial \mathcal{H}^T}{\partial \vec{u}} = 0 \quad (4.13)$$

However, if \vec{u} is constrained in set \mathcal{U} , one may use **Pontryagin's principle** to which replaces the previous equation with the requirement

$$\mathcal{H}(\vec{x}_{opt}(t), \vec{u}_{opt}(t), \vec{\lambda}_{opt}(t)) \leq \mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t)) \quad \forall t \in [t_0, t_F], \quad \vec{u} \in \mathcal{U} \quad (4.14)$$

For example, if one bounds the control simply by

$$\vec{u}_{min} \leq \vec{u} \leq \vec{u}_{max} \quad (4.15)$$

which requires for feasibility of the variation of \vec{u} at its limits which must only be in the direction allowed that one has

$$\begin{cases} \vec{u}_{opt}(t) = \vec{u}_{min} & \text{for } \frac{\partial \mathcal{H}}{\partial \vec{u}} \geq 0 \\ \vec{u}_{min} < \vec{u}_{opt}(t) < \vec{u}_{max} & \text{for } \frac{\partial \mathcal{H}}{\partial \vec{u}} = 0 \\ \vec{u}_{opt}(t) = \vec{u}_{max} & \text{for } \frac{\partial \mathcal{H}}{\partial \vec{u}} \leq 0 \end{cases} \quad (4.16)$$

It should be pointed out that Pontryagin's principle is a necessary condition, but only sufficient if $\mathcal{L}(\vec{x}(t), \vec{u}(t), t)$ and $f(\vec{x}(t), \vec{u}(t), t)$ are both convex in $\vec{x}(t)$ and $\vec{u}(t)$. However, a necessary and sufficient condition as an alternative to Pontryagin's principle, is given by the **Hamilton-Jacobi-Bellman (HJB) equation**

$$\begin{aligned}\frac{\partial \mathcal{V}_{opt}(\vec{x}, t)}{\partial t} &= -\frac{\partial \mathcal{H}}{\partial \vec{x}} \\ \mathcal{V}_{opt}(\vec{x}(t_f), t_f) &= \mathcal{E}(\vec{x}(t_f), t_f)\end{aligned}\quad (4.17)$$

where $\mathcal{V}(\vec{x}, t)$ is the continuous-time **cost-to-go**, also known as the **value function**, and replaces the costate or adjoint vector in the Hamiltonian expression. The **principle of optimality** states that regardless of the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision, i.e. for any $t > t_0$

$$\mathcal{V}_{opt}(\vec{x}, t) = \min_{\vec{u}(t) \forall t \in [t, t_f]} \mathcal{J}(\vec{x}, \vec{u}, t, t_f) \quad (4.18)$$

However, the HJB equation requires the value function has a well-defined gradient and, moreover, that this gradient can be further differentiated with respect to time. Thus, the second-order partial derivatives of \mathcal{V} must exist which may not be true in general. The discrete-time version of the HJB is the Bellman equation and forms the basis of dynamic programming which is introduced in Appendix B of this textbook.

Unconstrained LQ Optimal Control Problem

One of the fundamental OCPs is the **linear-quadratic OCP**, defined as having *linear* dynamics and a *quadratic* cost function/functional, with or without constraints. In particular, for continuous-time one has dynamics of the form

$$\dot{\vec{x}}(t) = A(t)x(t) + B(t)u(t) \quad (4.19)$$

and a cost functional of the form

$$\mathcal{J} = \frac{1}{2}x^T(t_f)Ex(t_f) + \frac{1}{2} \int_{t_0}^{t_f} \left(x^T(t)Q(t)x(t) + u^T(t)R(t)u(t) + 2x^T(t)S(t)u(t)dt \right) \quad (4.20)$$

This formulation contains **cost/weight matrices** where E is the **endpoint cost/weight matrix** or **terminal cost/weight matrix**, Q is the **state cost/weight matrix**, R is the **input cost/weight matrix**, and S is the **cross-cost/weight matrix**. The control designer selects the values of these matrices to balance the costs of large values of \vec{x} and \vec{u} . Furthermore, for the unconstrained case, E , Q , and R are all symmetric, positive semi-definite matrices so that the cost functional is non-negative. The controller which solves the LQ OCP is called the **linear-quadratic regulator (LQR)**, i.e. $\vec{u}^{\text{opt}}(t)$. The term **regulator** denotes that this controller steers the system state to 0. However, to generalize this recall the robust servomechanism control introduced in the last chapter. With the servomechanism to augment the dynamics, one can easily obtain the **linear-quadratic tracker (LQT)**. Furthermore, it should be noted that A/F , B/G , Q , R , and S can all vary with t .

The **unconstrained finite-horizon continuous-time LQ OCP** can be stated as

$$\begin{aligned} \vec{u}^{\text{opt}}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\text{argmin}} \quad \mathcal{J} = \vec{x}^T(t_f)E\vec{x}(t_f) + \int_0^{t_f} \vec{x}^T(t)Q\vec{x}(t) + \vec{u}^T(t)R\vec{u}(t) + 2\vec{x}^T(t)S\vec{u}(t)dt \\ &\text{subject to: } \dot{\vec{x}}(t) = A\vec{x}(t) + B\vec{u}(t) \\ &\text{initial condition: } \vec{x}(0) = \vec{x}_0 \end{aligned} \quad (4.21)$$

where the unconstrained finite-horizon continuous-time LQR is the optimal control function, $\vec{u}^{\text{opt}}(t)$, which minimizes the quadratic cost functional, J . Recalling the generalized calculus of variations solution method, one can assign the following Hamiltonian for this continuous-time OCP.

$$\mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t), t) = \vec{x}^T(t)Q\vec{x}(t) + \vec{u}^T(t)R\vec{u}(t) + 2\vec{x}^T(t)S\vec{u}(t) + \vec{\lambda}^T(A\vec{x}(t) + B\vec{u}(t)) \quad (4.22)$$

which allows one to define the unconstrained condition equations as

$$\begin{cases} \dot{\vec{\lambda}}(t) &= -\frac{\partial \mathcal{H}}{\partial \vec{x}} = -Q\vec{x} - S\vec{u} - A^T\vec{\lambda} \\ 0 &= \frac{\partial \mathcal{H}}{\partial \vec{u}} = \vec{u}^T R + \vec{x}^T S + \vec{\lambda}^T B \\ \vec{\lambda}(t_f) &= \frac{\partial \mathcal{E}(t_f)}{\partial \vec{x}}^T = \vec{x}^T(t_f)E \end{cases} \quad (4.23)$$

for the LQ OCP.

To solve this OCP, assume the costate has a linear form, i.e.

$$\vec{\lambda}(t) = P(t)\vec{x}(t) \quad (4.24)$$

where $P(t)$ is a symmetric matrix. Substituting this into the costate equations and including the state dynamics equation, one has

$$\begin{cases} \dot{\vec{x}} &= A\vec{x} + B\vec{u} \\ \frac{d}{dt}(P\vec{x}) = \dot{P}\vec{x} + P\dot{\vec{x}} &= -Q\vec{x} - S\vec{u} - A^T P\vec{x} \\ 0 &= \vec{u}^T R + \vec{x}^T S + \vec{x}^T P B \\ P(t_f)\vec{x}(t_f) &= E\vec{x}(t_f) \end{cases} \quad (4.25)$$

Next, substituting the first equation into the second equation results in

$$\dot{P}\vec{x} + PA\vec{x} + PB\vec{u} = -Q\vec{x} - S\vec{u} - A^T P\vec{x} \quad (4.26)$$

Then, rewriting the third equation, one has

$$\vec{u} = -R^{-1}(B^T P + S^T)\vec{x} \quad (4.27)$$

By substitution for \vec{u} into the newly derived equation, one has

$$\dot{P}\vec{x} + PA\vec{x} - PBR^{-1}(B^T P + S^T)\vec{x} = -Q\vec{x} + SR^{-1}(B^T P + S^T)\vec{x} - A^T P\vec{x} \quad (4.28)$$

By removing the common \vec{x} term and rearranging, one has

$$\dot{P} = -PA - A^T P + (PB + S)R^{-1}(B^T P + S^T) - Q \quad (4.29)$$

which is known as the **differential Riccati equation** and describes the dynamics of the continuous-time **Riccati matrix**, $P(t)$, and can be solved using the boundary condition on the costate, i.e.

$$P(t_f) = E \quad (4.30)$$

Thus, the Lagrangian multiplier problem has been reduced to a matrix-valued ODE which must be solved in reverse time from the end condition.

Finally, the **unconstrained finite-horizon continuous-time LQR** has the form

$$\vec{u}^{\text{opt}}(t) = -K(t)\vec{x}(t) \quad (4.31)$$

where

$$K(t) = R^{-1}(B^T P(t) + S^T) \quad (4.32)$$

which notably results in closed-loop state-space dynamics represented by

$$\dot{\vec{x}}(t) = (A - BK(t))\vec{x}(t) \quad (4.33)$$

Finally, if one considers the **unconstrained infinite-horizon continuous-time LQ OCP**, the solution is given by the steady-state of the Riccati differential equation, i.e.

$$0 = -PA - A^T P + (PB + S)R^{-1}(B^T P + S^T) - Q \quad (4.34)$$

which is known as the **continuous-time algebraic Riccati equation (CARE)**.

Infinite-Horizon Unconstrained Optimal Control

However, another approach to setting up an infinite-horizon OCP for this generalized feedback control system is to design a stabilizing LTI controller $K(s)$ to minimize the input-to-output gain from $\vec{d} \rightarrow \vec{e}$ by setting the cost functional $J(\vec{u})$ for LTI systems as some system norm $\|F_L(G, K)\|$, i.e.

$$\begin{aligned}\vec{u}_{opt}(s) &= K_{opt}(s) \vec{y}(s) = \underset{u(s)}{\operatorname{argmin}} \|F_L(G, K)\| \\ &\text{subject to:} \\ &\text{continuous dynamics: } \dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \\ &\text{constraints: } K \text{ stabilizing}\end{aligned}\tag{4.35}$$

where $F_L(G, K)$ defines a state-space representation as

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B_1\vec{d}(t) + B_2\vec{u}(t) \\ \vec{e}(t) &= C_1\vec{x}(t) + D_{12}\vec{u}(t) \\ \vec{y}(t) &= C_2\vec{x}(t) + D_{21}\vec{d}(t)\end{aligned}\tag{4.36}$$

This is often written more succinctly as

$$K_{opt}(s) = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \|F_L(G, K)\|\tag{4.37}$$

This chapter of the textbook considers two types of system norms, the \mathcal{H}_2 - and \mathcal{H}_∞ -norms, for this type of OCP.

4.2 Introductory Energy-, Time-, and Fuel-Optimal Control

Consider the **classical OCP**

$$\begin{aligned}\vec{u}_{opt}(t) &= \underset{u(t) \forall t \in [t_0, t_f]}{\operatorname{argmin}} \mathcal{J} = \mathcal{E}(\vec{x}(t_f), t_f) + \int_{t_0}^{t_f} \mathcal{L}(\vec{x}(t), \vec{u}(t), t) dt \\ &\text{subject to:} \\ &\text{continuous-time dynamics: } \dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \\ &\text{initial conditions: } \vec{x}(t_0) = \vec{x}_0 \\ &\text{control constraints: } |\vec{u}(t)| < u_{max}\end{aligned}\tag{4.38}$$

with t_0 fixed and t_f free where \mathcal{E} is the endpoint cost, \mathcal{L} is the running cost or Lagrangian, and $\mathcal{H} = \mathcal{L} + \vec{\lambda}^T f(\vec{x}(t), \vec{u}(t), t)$ is the corresponding Hamiltonian. This section will discuss some sub-types of OCPs that are applicable to many flight vehicle OCPs for guidance and control laws, namely, the energy, fuel, and time optimal control problems and will provide analysis of the solutions to these OCPs for LTI systems.

Minimum-Energy Optimal Control Problem

A **minimum-energy OCP** is defined as

$$\begin{aligned} \vec{u}_{opt}(t) = \underset{u(t) \forall t \in [0, t_f]}{\operatorname{argmin}} \quad \mathcal{J} &= \mathcal{E}(\vec{x}(t_f), t_f) + \int_0^{t_f} \vec{u}^T(t) \vec{u}(t) dt \\ \text{subject to: } \dot{\vec{x}}(t) &= f(\vec{x}(t), \vec{u}(t), t) \\ \text{initial conditions: } \vec{x}(t_0) - \vec{x}_0 &= 0 \\ \text{control constraints: } |\vec{u}(t)| &< 1 \end{aligned} \quad (4.39)$$

where the integral term is the \mathcal{L}_2 -norm of the control input signal. If one desires \vec{x}_f to be the origin and uses a quadratic terminal cost, i.e. $\mathcal{E}(\vec{x}(t_f), t_f) = \vec{x}^T(t_f) E \vec{x}(t_f)$, and linear dynamics, i.e. $f(\vec{x}(t), \vec{u}(t), t) = A(t) \vec{x} + B(t) \vec{u}$, then one has a special case of the LQR OCP with $Q = 0$ and $R = I$.

Assuming that (A, B) is controllable for all $t \in [0, t_f]$, the solution to the minimum-energy LQR OCP is given by the differential Riccati equation

$$\dot{P}(t) = -P(t)A(t) - A^T(t)P(t) + P(t)B(t)B^T(t)P(t) \quad (4.40)$$

with endpoint condition

$$P(t_f) = E \quad (4.41)$$

where the optimal control is given by

$$u_{opt} = -B^T(t)P(t)\vec{x}(t) \quad (4.42)$$

with closed-loop state-space dynamics represented by

$$\dot{\vec{x}}(t) = \left(A(t) - B(t)B^T(t)P(t) \right) \vec{x}(t) \quad (4.43)$$

It should be noted if one requires the system to converge to some $x(t_f) = x_f$, then one can define

$$E = \operatorname{diag}(\vec{E}) \quad (4.44)$$

and take the limit of the optimal control as this cost weight goes to ∞ , i.e.

$$\lim_{\vec{E} \rightarrow \infty} u_{opt}(E, t) \quad (4.45)$$

Furthermore, \vec{u} may also be constrained, e.g. $|\vec{u}_i| < u_{c,i} \forall i = 1, \dots, n_u$. This forms a constrained optimization problem which can be solved using Pontryagin's principle for admissible control set.

Minimum-Time Optimal Control Problem

A **minimum-time OCP** is defined as

$$\begin{aligned} \vec{u}_{opt}(t) = \underset{u(t) \forall t \in [0, t_f]}{\operatorname{argmin}} \quad \mathcal{J} &= \int_0^{t_f} dt = t_f \\ \text{subject to: } \dot{\vec{x}}(t) &= f(\vec{x}(t), \vec{u}(t), t) \\ \text{initial conditions: } \vec{x}(t_0) - \vec{x}_0 &= 0 \\ \text{final conditions: } \vec{x}(t_f) - \vec{x}_c &= 0 \\ \text{control constraints: } \vec{u}(t) &\in \mathcal{U} \end{aligned} \quad (4.46)$$

which corresponds to the Hamiltonian

$$\mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t), t) = 1 + \vec{\lambda}^T f(\vec{x}(t), \vec{u}(t), t) \quad (4.47)$$

This OCP generally requires use of Pontryagin's principle to solve as well as methods for determining admissible controls, $\vec{u}(t) \in \mathcal{U}$.

However, in this introductory section, consider linear dynamics, i.e. $f(\vec{x}(t), \vec{u}(t), t) = A(t)\vec{x} + B(t)\vec{u}$, a Hamiltonian

$$\mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t), t) = 1 + \vec{\lambda}^T A \vec{x} + \vec{\lambda}^T B \vec{u} \quad (4.48)$$

and a n_u -dimensional hypercube for independent control actuators as the admissible control set, i.e.

$$\mathcal{U} = \{\vec{u} \in \mathbb{R}^{n_u} : u_i \in [u_{i,\min}, u_{i,\max}], i = 1, \dots, n_u\} \quad (4.49)$$

Then, Pontryagin's principle states

$$\mathcal{H}(\vec{x}_{opt}(t), \vec{u}_{opt}(t), \vec{\lambda}_{opt}(t)) \leq \mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t)) \quad \forall t \in [0, t_f], \vec{u} \in \mathcal{U} \quad (4.50)$$

which can be simplified to

$$\vec{\lambda}_{opt}^T B \vec{u}_{opt} = \min_{\vec{u} \in \mathcal{U}} \vec{\lambda}^T B \vec{u} \quad \forall t \in [0, t_f] \quad (4.51)$$

Next, defining, \vec{b}_i as the i^{th} column of B and since each u_i can be chosen independently, one has

$$\sum_{i=1}^{n_u} \vec{\lambda}_{opt}^T b_i u_{i,opt} = \min_{u_{i,\min} \leq u_i \leq u_{i,\max}} \sum_{i=1}^{n_u} \vec{\lambda}^T \vec{b}_i u_i \quad \forall t \in [0, t_f] \quad (4.52)$$

Thus, one has the optimal control cases as

$$u_{i,opt}(t) = \begin{cases} u_{i,\max} & \text{if } \vec{\lambda}_{opt}^T \vec{b}_i < 0 \\ ? & \text{if } \vec{\lambda}_{opt}^T \vec{b}_i = 0 \\ u_{i,\min} & \text{if } \vec{\lambda}_{opt}^T \vec{b}_i > 0 \end{cases} \quad (4.53)$$

To solve for the ? element, recall that the costate equation is

$$\dot{\vec{\lambda}}_{opt}^T = -\frac{\partial \mathcal{H}}{\partial \vec{x}} = -A^T \vec{\lambda}_{opt} \quad (4.54)$$

or

$$\vec{\lambda}_{opt}(t) = \exp(A^T(t_{opt} - t)) \vec{\lambda}_{opt}(t_{opt}) \quad (4.55)$$

Thus, $\vec{\lambda}_{opt}^T \vec{b}_i = \vec{\lambda}_{opt}^T(t_{opt}) \exp(A^T(t_{opt} - t)) \vec{b}_i$ is only = 0 over some time interval if it's zero for all t and its derivatives, i.e.

$$\vec{\lambda}_{opt}^T(t_{opt}) \vec{b}_i = \vec{\lambda}_{opt}^T(t_{opt}) A \vec{b}_i = \dots = \vec{\lambda}_{opt}^T(t_{opt}) A^{n_x-1} \vec{b}_i \quad (4.56)$$

While $\vec{\lambda}_{opt}(t_{opt})$ cannot be 0 for $\vec{x}(t_0) \neq \vec{x}_0$, one requires each pair (A, \vec{b}_i) to be controllable also known as a **normal LTI system**.

Thus, \vec{u}_{opt} only takes values at the vertices of the hypercube \mathcal{U} and has a finite number of discontinuities, i.e. *switches*, between these values, and is unique. This result is known as the **bang-bang property** for linear systems. However, it can be shown that the assumption of normality for the bang-bang property of minimum time optimal control for a hypercube can be relaxed to a modified bang-bang principle for linear systems which states that while not every minimum-time optimal control is bang-bang, one can always select one that *is* bang-bang if \mathcal{U} is a convex polyhedron. Notably, one can also demonstrate a bang-bang property for normal linear-in-control systems.

The bang-bang property for LTI systems can be established without Pontryagin's principle by recalling the general solution to the LTI system as

$$\vec{x}(t) = \exp(A(t-t_0)) \vec{x}_0 + \int_{t_0}^{t_f} \exp(A(t-\tau)) B \vec{u}(\tau) d\tau \quad (4.57)$$

Then, for $t \geq t_0$, one has the *reachable set* from $\vec{x}(t_0) = \vec{x}_0$ at time t as

$$\mathcal{R}^t(\vec{x}_0) = \left\{ \exp(A(t-t_0)) \vec{x}_0 + \int_{t_0}^{t_f} \exp(A(t-\tau)) B \vec{u}(\tau) d\tau : \vec{u} \in \mathcal{U}, t_0 \leq \tau \leq t \right\} \quad (4.58)$$

where

$$t_{opt} = \operatorname{argmin}_t \vec{x}_{opt} \in \mathcal{R}^t(\vec{x}_0) \quad (4.59)$$

Thus, \vec{x}_{opt} must occur on the boundary of $\mathcal{R}^{t_{opt}}(\vec{x}_0)$ as if it was in the interior, one could reach it sooner. Noting that $\mathcal{R}^{t_{opt}}(\vec{x}_0)$ is compact and convex, there exists a hyperplane that passes through \vec{x}_{opt} and contains $\mathcal{R}^T(\vec{x}_0)$ on one side.

Choosing the normal vector to this hyperplane as $\vec{\lambda}_{opt}(t_{opt})$, one has

$$\vec{\lambda}_{opt}^T(t_{opt}) \vec{x}_{opt} \geq \vec{\lambda}_{opt}^T(t_{opt}) \vec{x} \quad \forall \vec{x} \in \mathcal{R}^{t_{opt}}(\vec{x}_0) \quad (4.60)$$

or, by definition

$$\begin{aligned} \int_{t_0}^{t_{opt}} \vec{\lambda}_{opt}^T(\tau) \exp(A(t-\tau)) B \vec{u}_{opt}(\tau) d\tau &\geq \int_{t_0}^{t_{opt}} \vec{\lambda}_{opt}^T(\tau) \exp(A(t-\tau)) B \vec{u}(\tau) d\tau \\ &\quad \forall \vec{u} \in \mathcal{U} \text{ from } [t_0, t_{opt}] \end{aligned} \quad (4.61)$$

and noting $\vec{\lambda}(\tau) = \exp(A^T(t_{opt}-\tau)) \vec{\lambda}_{opt}(t_{opt})$, one has

$$\int_{t_0}^{t_{opt}} \vec{\lambda}_{opt}^T(\tau) B \vec{u}_{opt}(\tau) d\tau \geq \int_{t_0}^{t_{opt}} \vec{\lambda}_{opt}^T(\tau) B \vec{u}(\tau) d\tau \quad \forall \vec{u} \in \mathcal{U} \text{ from } [t_0, t_{opt}] \quad (4.62)$$

from which one can show the optimal control cases as previously.

Minimum-Fuel Optimal Control Problem

A **minimum-fuel OCP** is defined as

$$\begin{aligned} \vec{u}_{opt}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\operatorname{argmin}} \quad \mathcal{J} = \int_0^{t_f} \sum_{i=1}^{n_u} |u_i(t)| dt \\ &\text{subject to: } \dot{\vec{x}}(t) = f(\vec{x}(t), \vec{u}(t), t) \\ &\text{initial conditions: } \vec{x}(t_0) - \vec{x}_0 = 0 \\ &\text{final conditions: } \vec{x}(t_f) - \vec{x}_c = 0 \\ &\text{control constraints: } |\vec{u}(t)| < 1 \end{aligned} \quad (4.63)$$

which corresponds to the Hamiltonian

$$\mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t), t) = \sum_{i=1}^{n_u} |u_i(t)| + \vec{\lambda}^T f(\vec{x}(t), \vec{u}(t), t) \quad (4.64)$$

The integral is the \mathcal{L}_1 -norm of the control input. Thus, a minimum-fuel optimal control is also known as \mathcal{L}_1 -optimal control. This OCP generally requires use of Pontryagin's principle to solve as well as methods for determining admissible controls, $\vec{u}(t) \in \mathcal{U}$.

However, in this introductory section, consider linear dynamics, i.e. $f(\vec{x}(t), \vec{u}(t), t) = A(t)\vec{x} + B(t)\vec{u}$, a Hamiltonian

$$\mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t), t) = \sum_{i=1}^{n_u} |u_i(t)| + \vec{\lambda}^T A \vec{x} + \vec{\lambda}^T B \vec{u} \quad (4.65)$$

and a n_u -dimensional hypercube for independent control actuators as the admissible control set, i.e.

$$\begin{aligned} \mathcal{U} &= \{ \vec{u} \in \mathbb{R}^{n_u} : u_i \in [u_{i,min}, u_{i,max}], i = 1, \dots, n_u \} \\ u_{i,min} &< 0 \\ u_{i,max} &> 0 \end{aligned} \quad (4.66)$$

Then, Pontryagin's principle states

$$\mathcal{H}(\vec{x}_{opt}(t), \vec{u}_{opt}(t), \vec{\lambda}_{opt}(t)) \leq \mathcal{H}(\vec{x}(t), \vec{u}(t), \vec{\lambda}(t)) \quad \forall t \in [0, t_f], \vec{u} \in \mathcal{U} \quad (4.67)$$

which can be simplified to

$$\sum_{i=1}^{n_u} |u_{i,opt}| + \vec{\lambda}_{opt}^T B \vec{u}_{opt} = \min_{\vec{u} \in \mathcal{U}} \sum_{i=1}^{n_u} |u_i| + \vec{\lambda}^T B \vec{u} \quad \forall t \in [0, t_f] \quad (4.68)$$

Next, defining, \vec{b}_i as the i^{th} column of B and since each u_i can be chosen independently, one has

$$\sum_{i=1}^{n_u} |u_{i,opt}| + \vec{\lambda}_{opt}^T b_i u_{i,opt} = \min_{u_{i,min} \leq |u_i| \leq u_{i,max}} \sum_{i=1}^{n_u} |u_i| + \vec{\lambda}^T \vec{b}_i u_i \quad \forall t \in [0, t_f] \quad (4.69)$$

Thus, one has the optimal control cases as

$$u_{i,opt}(t) = \begin{cases} u_{i,max} & \text{if } \vec{\lambda}_{opt}^T \vec{b}_i \leq -1 \\ 0 & \text{if } -1 \leq \vec{\lambda}_{opt}^T \vec{b}_i \leq 1 \\ u_{i,min} & \text{if } \vec{\lambda}_{opt}^T \vec{b}_i \geq 1 \end{cases} \quad (4.70)$$

This result is known as the **bang-off-bang property** for LTI systems as there may exist time intervals where $u_{i,opt}(t) = 0$ for the minimum-fuel OCP. However, in general, the exact time equation for \vec{u}_{opt} is not provided by the previous analysis as one does not directly know the dependence of the costate, $\vec{\lambda}$, on the state, \vec{x} , and there may be any number of minimum-fuel solutions depending on the state. Thus, minimum-fuel OCPs are more difficult to compute analytically although there are explicit solutions to second-order systems. Lastly, due to the similar nature of minimum-time and minimum-fuel, these two conditions can also be linearly combined into a minimum-time-fuel OCP.

Path Planning via Dubins Path

The planning system for flight vehicles typically solves two coupled problems, namely the mission plan and the path plan. The mission planning at the most general level consists of searching for unknown targets and/or optimally assigning targets, either as individual targets or sequences of targets, to all available flight vehicles in the mission. The selection or assignment of the targets or search paths for multiple flight vehicles typically involves the computation of a path cost for each vehicle, thus coupling the general mission planning problem with the path planning problem. In general, the optimal path can be solved via the **Euclidean shortest-path problem** which attempts to find the shortest-path within a continuously-valued Euclidean space, as opposed to a shortest-path problem based on discrete graphs which can use dynamic programming, e.g. Dijkstra's method, which naturally handles obstacles. For vehicles, the two-dimensional Euclidean shortest-path problem is also known as the **Dubins path problem** which can be solved in 2D and 3D using Pontryagin's principle with added constraints on the control.

For the two-dimensional Dubins path problem, one assumes the two-dimensional **Dubins vehicle** with dynamics

$$\vec{\dot{x}} = \begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\psi} \end{bmatrix} = f(\vec{x}, \vec{u}) = \begin{bmatrix} V \cos \psi \\ V \sin \psi \\ u \end{bmatrix} \quad (4.71)$$

where V is the *constant* velocity of the vehicle, (x, y) is the position of the vehicle, ψ is the heading angle, and u is the turn rate control input. The **instantaneous curvature** which is constrained by

$$\frac{-V}{R_{min}} \leq u \leq \frac{V}{R_{min}} \quad (4.72)$$

where R_{min} is the minimum radius of curvature. The objective for a Dubins path is starting at $t = 0$ at some state (x_0, y_0, ψ_0) , one wishes to achieve some other state, (x_f, y_f, ψ_f) , at time $t = t_f$ in the shortest amount of time.

Thus, the **2D Dubins path OCP** can be formally stated as

$$\begin{aligned}
 \vec{u}^{opt}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\operatorname{argmin}} J = \int_0^{t_f} dt = t_f \\
 &\text{subject to:} \\
 \text{dynamics} \quad &\left[\begin{array}{c} \dot{\vec{x}} \\ \dot{t} \end{array} \right] = \left[\begin{array}{c} f(\vec{x}, \vec{u}) \\ 1 \end{array} \right] \\
 \text{initial condition} \quad &\left[\begin{array}{c} x(0) \\ y(0) \\ \psi(0) \end{array} \right] = \left[\begin{array}{c} x_0 \\ y_0 \\ \psi_0 \end{array} \right] \\
 \text{final condition} \quad &\left[\begin{array}{c} x(t_f) \\ y(t_f) \\ \psi(t_f) \end{array} \right] = \left[\begin{array}{c} x_f \\ y_f \\ \psi_f \end{array} \right] \\
 \text{constraints} \quad &|u(t)| \leq \frac{V}{R_{min}}
 \end{aligned} \tag{4.73}$$

which corresponds to the Hamiltonian with $\vec{x}' = \begin{bmatrix} \vec{x} \\ t \end{bmatrix}$ as

$$\mathcal{H}(\vec{x}'(t), \vec{u}(t), \vec{\lambda}(t)) = \lambda^T f(\vec{x}'(t), \vec{u}(t), \vec{\lambda}(t)) \tag{4.74}$$

Then, the costate equations can be written as

$$\dot{\vec{\lambda}} = \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{\beta} \\ \dot{e} \end{bmatrix} = -\frac{\partial \mathcal{H}}{\partial \vec{x}'} = -\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ -V \sin \psi & V \cos \psi & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} p \\ q \\ \beta \\ e \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ pV \sin \psi - qV \cos \psi \\ 0 \end{bmatrix} \tag{4.75}$$

with final values for the costate vector as

$$\vec{\lambda}(t_f) = \begin{bmatrix} p(t_f) \\ q(t_f) \\ \beta(t_f) \\ e(t_f) \end{bmatrix} = \frac{\partial \mathcal{E}^T}{\partial \vec{x}'} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.76}$$

and with arbitrary, but not all zero, initial values.

Thus, p , q , and e are constant on $[0, t_f]$. Thus, by defining $r = \sqrt{p^2 + q^2} \geq 0$, one can write $p = r \cos \phi$ and $q = r \sin \phi$ which determines an angle $\phi \in [0, 2\pi)$ such that $\tan \phi = \frac{q}{p}$ which allows one to rewrite the third costate equation as

$$\dot{\beta} = rV \sin(\psi - \phi) \tag{4.77}$$

Furthermore, one has for the bounded control for the minimum-time OCP, $\frac{-V}{R_{min}} \leq u \leq \frac{V}{R_{min}}$, one requires that

$$\begin{cases} u(t) = \frac{-V}{R_{min}} & \text{for } \beta < 0 \\ u(t) = 0 & \text{for } \beta = 0 \\ u(t) = \frac{V}{R_{min}} & \text{for } \beta > 0 \end{cases} \tag{4.78}$$

Hence, if $\beta = 0$, then $\dot{\beta} = 0$ and $\psi = \phi$ or $\psi = \phi + \pi$ and the path is a line segment with direction ϕ . Otherwise if $\beta \neq 0$, then u must take its maximum or minimum values, i.e. $u = \pm \frac{V}{R_{\min}}$ and the path is an arc of circle of radius R . This demonstrates that any optimal path is the concatenation of arcs of circles of radius R and line segments all parallel to some fixed direction ϕ .

4.3 Semidefinite Programming in Optimal Control

Convexity

Many OCPs can be set up as convex optimization problems which uses the concept of convexity. To explain this concept, first consider two points in vector space \mathcal{V} , i.e. $\vec{v}_1 \in \mathcal{V}$ and $\vec{v}_2 \in \mathcal{V}$, then the line segment $L(\vec{v}_1, \vec{v}_2)$ between them contains the set of points defined by

$$L(\vec{v}_1, \vec{v}_2) = \{\vec{v} \in \mathcal{V} : \vec{v} = \mu \vec{v}_1 + (1 - \mu) \vec{v}_2 \text{ for some } \mu \in [0, 1]\} \quad (4.79)$$

Next, let $Q \subset \mathcal{V}$ be nonempty, then Q is a **convex set** of \mathcal{V} if for any $\vec{v}_1, \vec{v}_2 \in Q$, $L(\vec{v}_1, \vec{v}_2) \subset Q$. One can also consider the expression $\vec{v} = \mu \vec{v}_1 + (1 - \mu) \vec{v}_2$ as a weighed average, i.e. if $\mu_1, \mu_2 \in [0, 1]$ with $\mu_1 + \mu_2 = 1$, then $\vec{v} = \mu_1 \vec{v}_1 + \mu_2 \vec{v}_2$ which can be generalized to an average of n points as $\vec{v} = \mu_1 \vec{v}_1 + \dots + \mu_n \vec{v}_n$ with $\mu_1, \dots, \mu_n \in [0, 1]$ and $\mu_1 + \dots + \mu_n = 1$.

Furthermore, the set comprised of all weighted averages of the points $\vec{v}_1, \dots, \vec{v}_n$ is called its convex hull, $co(\{\vec{v}_1, \dots, \vec{v}_n\})$. More generally, given a set Q one can define its **convex hull**, $co(Q)$, by the set $\{\vec{v} \in \mathcal{V} : \text{there exists } n \text{ and } \vec{v}_1, \dots, \vec{v}_n \in Q \text{ such that } \vec{v} \in co(\{\vec{v}_1, \dots, \vec{v}_n\})\}$, i.e. the convex hull of Q is the collection of all possible weighted averages of points in Q . Furthermore, one can show

- the subset $Q \subset co(Q)$ is satisfied
- the convex hull $co(Q)$ is convex
- $co(Q) = co(co(Q))$
- a set Q is convex if and only if $co(Q) = Q$

where it should also be noted by definition, the intersection of convex sets is always convex.

Lastly, a set $Q \subset \mathcal{V}$ is called a **cone** if it is closed under positive scalar multiplication, i.e. if

$$\vec{v} \in Q \text{ implies } t \vec{v} \in Q \text{ for every } t > 0 \quad (4.80)$$

thus, subspaces are cones, but cones are a broader set, e.g. the half-line $C_v = \{\alpha \vec{v} : \alpha > 0\}$ for a fixed vector \vec{v} . Furthermore, a **convex cone** are cones closed under addition, i.e. $\vec{v}_1, \vec{v}_2 \in Q$ implies $\vec{v}_1 + \vec{v}_2 \in Q$.

Semidefinite Programming

Many generalized OCPs can be formulated as **semidefinite programs (SDP)**, a type of convex optimization which use LMI constraints on the solution space X , e.g. ARIs. The general form of an SDP can be formulated

as

$$\begin{aligned} X^{opt} = \underset{X \in \mathcal{X}}{\operatorname{argmin}} \quad & c(X) \\ \text{subject to: } & F(X) \leq Q \\ & X \in \mathcal{X} \end{aligned} \tag{4.81}$$

where $c(X)$ is a *linear* functional on the vector space \mathcal{X} , and is also known as a type of **linear objective problem**.

Note that for this SDP, the LMI constraint must be feasible. Thus, the simplest SDP may be formulated as a **feasibility problem** if there exists $X \in \mathcal{X}$ satisfying $F(X) \leq Q$ which can also be recast as the linear objective problem

$$\begin{aligned} J = \inf \quad & t \\ \text{subject to: } & F(X) - tI \leq Q \end{aligned} \tag{4.82}$$

where if and only if $J < 0$, then the LMI $F(X) \leq Q$ is feasible. Thus, the feasibility question can be made a part of the SDP and the focus of solving SDPs is on solving linear objective problems with LMI constraints.

To derive solutions for SDPs, note that the set $C = \{X \in \mathcal{X} : F(X) \leq Q\}$ is a **convex set** in \mathcal{X} . As a proof, let $X_1, X_2 \in C$ which means one satisfies $F(X_1) \leq Q$ and $F(X_2) \leq Q$. Then, considering any point $X_3 \in L(X_1, X_2)$, i.e. $X_3 = \mu X_1 + (1 - \mu)X_2$ for some value $\mu \in [0, 1]$, using the linearity of $F()$, one has

$$F(X_3) = \mu F(X_1) + (1 - \mu)F(X_2) \leq \mu Q + (1 - \mu)Q = Q \tag{4.83}$$

where the inequality follows from the fact that positive definite matrices are convex cones. Therefore, $X_3 \in C$.

This property generalizes to any convex optimization problem and ensures these problems can be solved globally, not just locally, by numerical search algorithms. At first glance, one may think that the minimum if it exists must lie on the boundary of the feasible set which would restrict the search to the boundary which is the case for **linear programming**. However, the boundary in general SDPs is complicated, thus, **interior point methods** are favored in practice. As an example convex optimization method, suppose one currently has the point X_n in the feasible set. An immediate consequence is that one only needs to keep the set

$$\{X \in C : c(X) \leq c(X_n)\} \tag{4.84}$$

for the remaining search for the global minimum. This amounts to setting C with a half-space, thus one can progressively shrink the feasibility region to zero, provided one is able to successively generate a “good” feasible point, e.g. X_{n+1} .

Many optimization methods are based on this principle with one of the simplest known as the **ellipsoid algorithm** that alternates between “cutting” and bounding the resulting set by an ellipsoid where X_{n+1} would be the center of such ellipsoid. More efficient methods for SDPs are based on **barrier functions** to impose the feasibility constraint. Here the idea is to minimize the function

$$c(X) + \alpha\phi(X) \tag{4.85}$$

where $\alpha > 0$ and the barrier function $\phi(X)$ is convex and approaches infinity on the boundary of the feasible set, e.g. for the set C

$$\phi(X) = -\log(\det[Q - F(X)]) \quad (4.86)$$

can serve as a barrier function. Thus, provided one starts from a feasible point, the minimization can be globally solved by unconstrained optimization methods, e.g. Newton's method. By successively reducing the weight of the barrier function, an iteration is produced which can be shown to converge to the global minimum with computational complexity of polynomial growth with problem size characterized by the dimension of X and constraint set \mathbb{H}^n . Additional details on convex optimization algorithms are beyond the scope of this textbook.

Bounded Real Lemma

As will be shown, solutions to infinite-horizon OCPs can be found by setting up algebraic Riccati equations (ARE) of the general form

$$A^T P + PA + Q + PRP = 0 \quad (4.87)$$

where typically, one is required to find a particular solution $P = P^T \in \mathbb{R}^{n_x \times n_x}$ such that $A + RP$ is stable, i.e. all its eigenvalues have strict negative real part. For such problems, one can define the Hamiltonian matrix of the ARE of (A, Q, R) as

$$H = \begin{bmatrix} A & R \\ -Q & -A^T \end{bmatrix} \quad (4.88)$$

Thus, the ARE becomes

$$\begin{bmatrix} P & -I \end{bmatrix} H \begin{bmatrix} I \\ P \end{bmatrix} = 0 \quad (4.89)$$

where the eigenvalue decomposition of H is key to solving the ARE. In this case, one can state that if A , $Q = Q^T$ and $R = R^T$ are given, H has no purely imaginary axis eigenvalues, and $R \geq 0$ or $R \leq 0$, and (A, R) is stabilizable, then the ARE of (A, Q, R) has a unique solution $P = P^T$ such that $A + RP$ is stable.

However, for more advanced control synthesis, one can instead use **algebraic Riccati inequality (ARI)** which is defined as

$$A^T P + PA + Q + PRP < 0 \quad (4.90)$$

which is a type of linear matrix inequality which will be discussed in the following section. For example, consider the inequality

$$A^T P + PA + C^T C + \gamma^{-2} PBB^T P < 0 \quad (4.91)$$

which is quadratic in $P \in \mathbb{S}^n > 0$. However, rewriting this as

$$\begin{bmatrix} A^T P + PA + C^T C + \gamma^{-2} PBB^T P & 0 \\ 0 & -\gamma^{-2} I \end{bmatrix} < 0 \quad (4.92)$$

by the Schur Complement Lemma, one can form the equivalent LMI for P .

$$\begin{bmatrix} A^T P + PA + C^T C & PB \\ B^T P & -\gamma^2 I \end{bmatrix} < 0 \quad (4.93)$$

or

$$\begin{bmatrix} A^T P + PA & PB \\ B^T P & \gamma^{-2} I \end{bmatrix} < \begin{bmatrix} -C^T C & 0 \\ 0 & 0 \end{bmatrix} \quad (4.94)$$

which is an LMI in P as the left side can be assigned as $F(P) : \mathbb{S}^n \rightarrow \mathbb{S}^n$ and the right side can be assigned as $Q \in \mathbb{S}$.

In this case, one can relate solutions to AREs and ARIs for OCPs. One of the important lemmas between AREs/ARIs is known as the **Bounded Real Lemma**, also known as the **Kalman-Yacubovich-Popov (KYP) lemma**, which can be stated as follows. Consider the following LTI system, $F_L(G, K)$:

$$\begin{aligned} \dot{\vec{x}}(t) &= A_L \vec{x}(t) + B_1 \vec{d}(t) \\ \vec{e}(t) &= C_1 \vec{x}(t) \end{aligned} \quad (4.95)$$

and let $\gamma > 0$ be given. Then, the following three statements are equivalent.

1. $F_L(G, K)$ is stable, i.e. A_L is stable, and $\|F_L(G, K)\|_\infty < \gamma^2$
2. There exists a unique $P_1 \geq 0$ such that $A_L + \gamma^{-2} P_1 B_1 B_1^T P_1$ is stable and satisfies the ARE:

$$A^T P_1 + P_1 A + C_1^T C_1 + \gamma^{-2} P_1 B_1 B_1^T P_1 = 0 \quad (4.96)$$

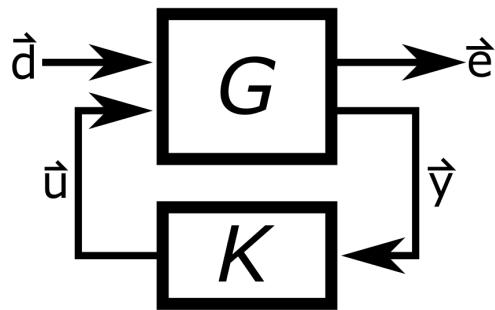
3. There exists $P_2 > 0$ satisfying the strict ARI:

$$A^T P_2 + P_2 A + C_1^T C_1 + \gamma^{-2} P_2 B_1 B_1^T P_2 < 0 \quad (4.97)$$

where to solve ARIs for P_2 requires **semidefinite programming (SDP)**, a class of convex optimization, which will be discussed in the following subsection and then applied to both the \mathcal{H}_2 and \mathcal{H}_∞ OCPS in later sections.

Stabilizing State Feedback as LMI

Consider the generalized feedback control system



with generalized plant, $G(s)$:

$$\begin{aligned} \dot{\vec{x}}(t) &= A \vec{x}(t) + B_1 \vec{d}(t) + B_2 \vec{u}(t) \\ \vec{e}(t) &= C_1 \vec{x}(t) + D_{11} \vec{d}(t) + D_{12} \vec{u}(t) \end{aligned} \quad (4.98)$$

and a state feedback control policy for $K(s)$ as

$$\vec{u}(t) = D_K \vec{x}(t) \quad (4.99)$$

which results in a closed-loop system

$$\begin{aligned} \dot{\vec{x}}(t) &= (A + B_2 D_K) \vec{x}(t) + B_1 \vec{d}(t) \\ \vec{e}(t) &= (C_1 + D_{12} D_K) \vec{x}(t) + D_{11} \vec{d}(t) \end{aligned} \quad (4.100)$$

which is stable if and only if $A_L = A + B_2 D_K$ is stable.

To construct an LMI for constructing a stabilizing D_K , one can enforce this stable condition by a matrix Lyapunov inequality

$$A_L P + P A_L^T < 0 \quad (4.101)$$

which has a solution $P > 0$. Substituting for A_L , one has

$$(A + B_2 D_K) P + P (A + B_2 D_K)^T < 0 \quad (4.102)$$

which can be expanded as

$$AP + PA^T + B_2(D_K P) + (PD_K^T)B_2^T < 0 \quad (4.103)$$

Then, substituting $Y = D_K P$, then one has the LMI in P and Y as

$$[A \quad B_2] \begin{bmatrix} P \\ Y \end{bmatrix} - [X \quad Y^T] \begin{bmatrix} A^T \\ B_2^T \end{bmatrix} < 0 \quad (4.104)$$

which can be solved for P and Y and one can obtain the stabilizing feedback controller as $D_K = YP^{-1}$. It should be noted that this process can be expanded to general LMI characterizations for non-state feedback controllers D_K .

4.4 Introductory \mathcal{H}_2 Optimal Control

Unconstrained Continuous-Time \mathcal{H}_2 OCP

The objective of \mathcal{H}_2 optimal control is to find K that minimizes the \mathcal{H}_2 -norm of the generalized feedback control system, i.e.

$$K^{opt} = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \|F_L(G, K)\|_2 \quad (4.105)$$

where $F_L(G, K)$ defines a state-space model as

$$\begin{aligned} \dot{\vec{x}}(t) &= A \vec{x}(t) + B_1 \vec{d}(t) + B_2 \vec{u}(t) \\ \vec{e}(t) &= C_1 \vec{x}(t) + D_{12} \vec{u}(t) \\ \vec{y}(t) &= C_2 \vec{x}(t) + D_{21} \vec{d}(t) \end{aligned} \quad (4.106)$$

where $D_{11} = 0$ for a finite \mathcal{H}_2 -norm and $D_{22} = 0$ without loss of generality for ease of notation. Recall that for a solution to the \mathcal{H}_2 OCP to exist, (A, B_2) must be stabilizable and (A, C_2) must be detectable.

There are two groups to which all \mathcal{H}_2 OCPs can be divided, regular and singular. A **regular \mathcal{H}_2 OCP** additionally assumes that (1) for the LTI plant state-space model (A, B_2, C_1, D_{12}) there are no zeros on the $j\omega$ axis and $(D_{12}^T D_{12})^{-1}$ exists, i.e. injective, and (2) for the LTI plant state-space model (A, B_1, C_2, D_{21}) there are no zeros on the $j\omega$ axis and $(D_{21} D_{21}^T)^{-1}$ exists, i.e. surjective. For regular \mathcal{H}_∞ OCPs one can find a *unique* solution, whereas a sub-optimal solution must be found for singular \mathcal{H}_2 OCPs using a modification of the semidefinite programming approach shown here.

Furthermore, note that by definition of the \mathcal{H}_2 -norm in the time domain with $\vec{d}(t)$ constrained as impulse inputs, one can alternatively state the \mathcal{H}_2 cost function as

$$\mathcal{J}(\vec{x}, \vec{u}) = \frac{1}{2} \int_0^\infty \vec{e}^T \vec{e} d\tau \quad (4.107)$$

and by substitution, one has

$$\mathcal{J}(\vec{x}, \vec{u}) = \frac{1}{2} \int_0^\infty (C_1 \vec{x}(t) + D_{12} \vec{u}(t))^T (C_1 \vec{x}(t) + D_{12} \vec{u}(t)) d\tau \quad (4.108)$$

$$\mathcal{J}(\vec{x}, \vec{u}) = \frac{1}{2} \int_0^\infty \vec{x}(t)^T C_1^T C_1 \vec{x}(t) + 2 \vec{x}(t)^T C_1^T D_{12} \vec{u}(t) + \vec{u}(t)^T D_{12}^T D_{12} \vec{u}(t) d\tau \quad (4.109)$$

which is simply a more generalized form of the infinite-horizon continuous-time LQ OCP with $Q = C_1^T C_1$, $S = C_1^T D_{12}$, and $R = D_{12}^T D_{12}$.

Thus, the infinite-horizon LQR OCP is the state feedback \mathcal{H}_2 OCP, i.e. $C_2 = I$, $B_1 = \text{diag}(\vec{x}_0)$ and $D_{21} = 0$. This chapter of the textbook focuses on the state feedback \mathcal{H}_2 OCP for which one can simplify the problem to

$$\vec{y}(t) = \vec{x} \quad (4.110)$$

and

$$K : \vec{u} = D_k \vec{x} \quad (4.111)$$

and is always a regular \mathcal{H}_2 OCP. However, output feedback \mathcal{H}_2 OCPs may be regular or singular. However, the observer feedback \mathcal{H}_2 OCP as a regular type allows one to apply the separation principle to the problem to obtain the LQR and the **linear-quadratic estimator (LQE)**. Also, it should be noted that when \vec{d} are modeled as Gaussian noise, this type of problem is also known as the **linear-quadratic-Gaussian (LQG) OCP**, a fundamental OCP for stochastic dynamical systems, and is addressed in a subsequent chapters.

ARE Synthesis for State Feedback \mathcal{H}_2 OCP

Recall the solution to the finite-horizon continuous-time LQR OCP used the **Riccati differential equation**, i.e.

$$\dot{P} = -PA - A^T P + (PB + S) R^{-1} (B^T P + S^T) - Q \quad (4.112)$$

Furthermore, if one considers the **unconstrained infinite-horizon continuous-time LQ OCP** which sets $t_f = \infty$, the **unconstrained infinite-horizon continuous-time LQR** requires the steady-state solution of the Riccati differential equation, i.e.

$$0 = PA + A^T P - (PB + S) R^{-1} (B^T P + S^T) + Q \quad (4.113)$$

which is also known as the **continuous algebraic Riccati equation (CARE)** which can be solved using standard algorithms. Thus, the optimal control is given by

$$\vec{u}(t) = D_K \vec{x}(t) = -R^{-1}(B^T P + S^T) \vec{x}(t) \quad (4.114)$$

which notably results in closed-loop dynamics represented by

$$\dot{\vec{x}}(t) = (A + BD_K) \vec{x}(t) \quad (4.115)$$

SDP Synthesis for State Feedback \mathcal{H}_2 OCP

To setup the \mathcal{H}_2 OCP as a SDP, one requires the following result. Consider a strictly proper state-space LTI system, G , i.e. $D = 0$. Then, the state matrix, A , is stable and $\|G\|_2 < 1$ if and only if, there exists a $W \in \mathbb{S}^{n_x}$ such that

$$\text{Tr}[CWC^T] < 1 \quad (4.116)$$

and

$$AW + WA^T + BB^T < 0 \quad (4.117)$$

As a necessary proof, assume A is stable and $\|G\|_2 < 1$, then, by definition of the \mathcal{H}_2 -norm in terms of the controllability gramian W_C , one has

$$\|G\|_2 = \text{Tr}[CWC^T] < 1 \quad (4.118)$$

Next, consider the perturbed expression for $\epsilon > 0$

$$W(\epsilon) = \int_0^\infty e^{At} (BB^T + \epsilon I) e^{At} dt \quad (4.119)$$

where it should be noted that $W(\epsilon) > 0$ for $\epsilon > 0$. Furthermore, as $W(\epsilon)$ is a continuous function of ϵ and $W(\epsilon) = W_C$ for $\epsilon = 0$. By continuity, one has $\text{Tr}[CWC^T] < 1$ for some $\epsilon > 0$. Thus, this matrix satisfies the matrix Lyapunov equation

$$AW + WA^T + (BB^T + \epsilon I) = 0 \quad (4.120)$$

or

$$AW + WA^T + BB^T = -\epsilon I < 0 \quad (4.121)$$

As a sufficient proof, assume $W > 0$ such that

$$\text{Tr}[CWC^T] < 1 \quad (4.122)$$

and

$$AW + WA^T + BB^T < 0 \quad (4.123)$$

Thus, as $BB^T > 0$, $AW + WA^T < 0$ so that A is stable. Recalling the controllability gramian satisfies

$$AW_C + W_C A^T + BB^T = 0 \quad (4.124)$$

and using a technical result that solutions of the Lyapunov equality lower bound solutions of the corresponding Lyapunov inequality, one has

$$W_C \leq W \quad (4.125)$$

which implies

$$\text{Tr}[CW_C C^T] \leq \text{Tr}[CWC^T] < 1 \quad (4.126)$$

or, by definition

$$\|G\|_2 < 1 \quad (4.127)$$

Thus, by the previous \mathcal{H}_2 -norm and stability lemma with an additional scaling argument with γ , one can state that there exists a $D_K \in \mathbb{R}^{n_u \times n_y}$ such that A_L is stable and satisfies $\|F_L(G, K)\|_2 < \gamma$ if and only if there exists $P > 0$ and D_K such that

$$A_L P + PA_L^T + B_L B_L^T < 0 \quad (4.128)$$

and

$$\text{Tr}[C_L P C_L^T] < \gamma^2 \quad (4.129)$$

Substituting for the closed-loop matrices, one has

$$(A + B_2 D_K)P + P(A + B_2 D_K)^T + B_1 B_1^T < 0 \quad (4.130)$$

and

$$\text{Tr}[(C_1 + D_{12} D_K)P(C_1 + D_{12} D_K)^T] < \gamma^2 \quad (4.131)$$

which is a nonlinear matrix inequality in (P, D_K) due to the bilinear terms.

However, one can define a change of variable from D_K to

$$Q = D_K P \quad (4.132)$$

which allows one to write

$$\begin{bmatrix} A & B_2 \end{bmatrix} \begin{bmatrix} P \\ Q \end{bmatrix} + \begin{bmatrix} P & Q \end{bmatrix} \begin{bmatrix} A^T \\ B_2^T \end{bmatrix} + B_1 B_1^T < 0 \quad (4.133)$$

which is now an LMI in variables (P, Q) , however

$$\text{Tr}[(C_1 + D_{12} Q P^{-1})P(C_1 + D_{12} Q P^{-1})^T] < \gamma^2 \quad (4.134)$$

or

$$\text{Tr}[(C_1 P + D_{12} Q)P^{-1}(C_1 P + D_{12} Q)^T] < \gamma^2 \quad (4.135)$$

is not a convex condition, but one can convexify this inequality by introducing a **slack variable**, $R \in \mathbb{S}^{n_e}$, based on the following linear algebra fact: if $M_1 \leq M_2$, then $\text{Tr}[M_1] \leq \text{Tr}[M_2]$.

In this case, one can write the trace inequality above as two separate inequalities

$$\begin{aligned} (C_1 P + D_{12} Q)P^{-1}(C_1 P + D_{12} Q)^T &< R \\ \text{Tr}[R] &< \gamma^2 \end{aligned} \quad (4.136)$$

Then, by subtracting R from both sides and using the Schur complement lemma, one has

$$\begin{bmatrix} R & C_1 P + D_{12} Q \\ (C_1 P + D_{12} Q)^T & P \end{bmatrix} > 0 \quad \text{Tr}[R] < \gamma^2 \quad (4.137)$$

Thus, one can state the \mathcal{H}_2 OCP as an SDP in (P, Q, R, γ) , i.e.

$$\begin{aligned} (P, Q, R, \gamma)^{opt} = & \underset{P \in \mathbb{S}^{nx}, Q \in \mathbb{R}^{nu \times nx}, R \in \mathbb{S}^{ne}, \gamma > 0}{\operatorname{argmin}} \gamma \\ \text{subject to: } & [A \ B_2] \begin{bmatrix} P \\ Q \end{bmatrix} + [P \ Q] \begin{bmatrix} A^T \\ B_2^T \end{bmatrix} + B_1 B_1^T < 0 \\ & - \begin{bmatrix} R & C_1 P + D_{12} Q \\ (C_1 P + D_{12} Q)^T & P \end{bmatrix} < 0 \\ & \text{Tr}[R] - \gamma^2 < 0 \end{aligned} \quad (4.138)$$

with the \mathcal{H}_2 optimal controller, K^{opt} , as $\vec{u}(t) = D_K \vec{x}(t)$ reconstructed with

$$D_K = Q P^{-1} \quad (4.139)$$

Cost Matrix Design for \mathcal{H}_2 Control

Control designers using the LQR method must select the Q , R , and S cost matrices which implicitly affect the optimal controller. Thus, any implementation using the LQR method should use some useful guidelines. First, recall that one requires Q and R to be symmetric and positive semi-definite for the Lagrangian to be non-negative. Secondly, for an easier design process, one typically sets $S = 0$ and uses one of the four following methods for selecting Q and R .

The first method uses a relative cost, ρ , between the state and input which sets

$$Q = I \quad R = \rho^2 I \quad (4.140)$$

where the cost functional has become

$$\mathcal{J} = \int_0^\infty \vec{x}^T \vec{x} + \rho^2 \vec{u}^T \vec{u} \quad (4.141)$$

which can be said to balance the \mathcal{L}_2 -norm of the state and input through ρ . By varying ρ from $0 \rightarrow \infty$, one can sweep through the different values of ρ to find a satisfactory response. One analysis plot using this sweeping method is ρ versus \mathcal{J} which is called the **optimal tradeoff curve** to identify the lowest cost overall. Another analysis plot would be a root locus as a function of ρ .

The second method uses a relative cost, ρ , between output and input which sets

$$Q = C^T C \quad R = \rho^2 I \quad (4.142)$$

where one has incorporated an output equation with no feedthrough term, $\vec{y} = C \vec{x}$, in order to balance the \mathcal{L}_2 -norm of the output with the input. With this method, the cost functional has become

$$\mathcal{J} = \int_0^\infty \vec{y}^T \vec{y} + \rho^2 \vec{u}^T \vec{u} \quad (4.143)$$

which can be said to balance the \mathcal{L}_2 -norm of output and input through ρ . Here one can also vary ρ from $0 \rightarrow \infty$ to find satisfactory response from the possible options.

The third method uses individual diagonal costs in addition to the relative cost, ρ , which sets

$$Q = \begin{bmatrix} q_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & q_n \end{bmatrix} \quad R = \rho^2 \begin{bmatrix} r_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & r_p \end{bmatrix} \quad (4.144)$$

where each q_i and r_i are selected to normalize the state and input for “equal” levels of error or effort, respectively. This can also be thought of as a weighted \mathcal{L}_2 -norm of the state and input, and can also be extended for a weighted \mathcal{L}_2 -norm of the output and input using the no-feedthrough model, $\vec{y} = C\vec{x}$. It should be noted that the normalization is typically based on units. As an example, consider that an error of 5 m/s in x_1 is *as poor as* an error of 3° in x_2 . Then, by setting

$$q_1 = \left(\frac{1}{5}\right)^2 \quad (4.145)$$

and

$$q_2 = \left(\frac{1}{3}\right)^2 \quad (4.146)$$

one has normalized $q_1 x_1^2 = 1$ and $q_2 x_2^2 = 1$ for comparable levels of error. Here again, one can vary ρ from $0 \rightarrow \infty$ to find a satisfactory response. Here, choosing the diagonal costs may require additional trial and error if no simple method gives a satisfactory controller and is the primary task for the control designer.

The fourth method uses **Parseval’s theorem** which converts scalar quadratic functions in the time domain to the frequency domain using Fourier transforms, i.e.

$$\mathcal{J} = \int_0^\infty \vec{x}^T(t) Q \vec{x}(t) + \vec{u}^T(t) R \vec{u}(t) dt = \frac{1}{2\pi} \int_{-\infty}^\infty \vec{x}^T(-j\omega) Q \vec{x}(j\omega) + \vec{u}^T(-j\omega) R \vec{u}(j\omega) d\omega \quad (4.147)$$

where $\vec{x}(j\omega)$ is the continuous-time Fourier transform of $\vec{x}(t)$ and $\vec{u}(j\omega)$ is the continuous-time Fourier transform of $\vec{u}(t)$. Here, the matrices Q and R can be made into functions of frequency, ω , and forms a sort of loop-shaping design method.

In any of these cases, the Q and R matrices of the quadratic cost functional can be related to the eigenvalues of the closed-loop state dynamics for the \mathcal{H}_2 controller, i.e.

$$\dot{\vec{x}} = (A + BD_K) \vec{x} = A_L \vec{x} \quad (4.148)$$

by considering the Hamiltonian matrix, H , given for continuous-time by

$$H = \begin{bmatrix} A & -BR^{-1}B^T \\ -Q & -A^T \end{bmatrix} \quad (4.149)$$

which contains $2n_x$ eigenvalues. n_x of these are the eigenvalues of A_L with negative real parts and n_x have positive real parts, unstable, but stable backward in time. To show, recall the characteristic equation of the closed-loop whose zeros are the eigenvalues can be defined as

$$\phi_{cl} = \det [sI - A - BD_K] \quad (4.150)$$

then, by the above fact, one has

$$\phi_{cl}(s)\phi_{cl}(-s) = \det[sI - H] \quad (4.151)$$

$$\phi_{cl}(s)\phi_{cl}(-s) = \det \begin{bmatrix} sI - A & BR^{-1}B^T \\ Q & sI + A^T \end{bmatrix} \quad (4.152)$$

$$\phi_{cl}(s)\phi_{cl}(-s) = \det \begin{bmatrix} sI - A & BR^{-1}B^T \\ 0 & (sI + A^T) - Q(sI - A)^{-1}BR^{-1}B^T \end{bmatrix} \quad (4.153)$$

and multiplying the first row by $-Q(sI - A)^{-1}$ and adding to the second row, one has

$$\phi_{cl}(s)\phi_{cl}(-s) = \det[sI - A] \det[(sI + A^T) - Q(sI - A)^{-1}BR^{-1}B^T] \quad (4.154)$$

$$\phi_{cl}(s)\phi_{cl}(-s) = \phi(s) \det \left[(sI + A^T) \left(I - (sI + A^T)^{-1}Q(sI - A)^{-1}BR^{-1}B^T \right) \right] \quad (4.155)$$

$$\phi_{cl}(s)\phi_{cl}(-s) = \phi(s) \det[(sI + A^T)] \det[I - (sI + A^T)^{-1}Q(sI - A)^{-1}BR^{-1}B^T] \quad (4.156)$$

Next, factoring $Q = Q_1^T Q_1$ and $BR^{-1}B^T = R_1 R_1^T$, one has

$$\phi_{cl}(s)\phi_{cl}(-s) = \phi(s)\phi(-s) \det[I - (sI + A^T)^{-1}Q_1^T Q_1 (sI - A)^{-1}R_1 R_1^T] \quad (4.157)$$

and using the identity, $\det[I - AB] = \det[I - BA]$, one has

$$\phi_{cl}(s)\phi_{cl}(-s) = \phi(s)\phi(-s) \det[I - Q_1(sI - A)^{-1}R_1 R_1^T (sI + A^T)^{-1}Q_1^T] \quad (4.158)$$

Then, letting $H_1 = Q_1(sI - A)^{-1}R_1$, one has

$$\phi_{cl}(s)\phi_{cl}(-s) = \phi(s)\phi(-s) \det[I - H_1(s)H_1^T(-s)] \quad (4.159)$$

Finally, considering a cost functional with relative cost, ρ^2

$$\mathcal{J} = \int_0^\infty \vec{x}^T Q \vec{x} + \rho^2 \vec{u}^T R \vec{u} \quad (4.160)$$

one can show that the poles of the Hamiltonian, i.e. the zeros of $\phi(s)\phi(-s) \det[I - H_1(s)H_1^T(-s)]$, are also the zeros of $\phi(s)\phi(-s) \det[\rho I - H_1(s)H_1^T(-s)]$ which show that as $\rho \rightarrow 0$, some of the roots will go to infinity and those that stay finite approach the zeros of $H_1(s)$ and these finite zeros dictate the dynamic response of the LQR. As $\rho \rightarrow \infty$, the roots of $\phi_{cl}(s)$ are the n_x stable roots of $\phi(s)\phi(-s)$. Hence, one can see that shaping the zeros of $H_1(s)$ is crucial to the design of the LQR controller.

4.5 Introductory \mathcal{H}_∞ Optimal Control

Unconstrained Continuous-Time \mathcal{H}_∞ OCP

The design goal of \mathcal{H}_∞ optimal control is to find a stabilizing K that minimizes the \mathcal{H}_∞ -norm of the generalized feedback control system, i.e.

$$K^{opt} = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \|F_L(G, K)\|_\infty \quad (4.161)$$

where $F_L(G, K)$ defines a state-space model as

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B_1\vec{d}(t) + B_2\vec{u}(t) \\ \vec{e}(t) &= C_1\vec{x}(t) + D_{11}\vec{d}(t) + D_{12}\vec{u}(t) \\ \vec{y}(t) &= C_2\vec{x}(t) + D_{21}\vec{d}(t) + D_{22}\vec{u}(t)\end{aligned}\quad (4.162)$$

Recalling the definition of the \mathcal{H}_∞ -norm, one can recast this OCP in a min-max OCP framework as

$$K^{opt}, \vec{d}^{opt} = \underset{\substack{K \text{ stabilizing} \\ 0 \neq \|\vec{d}\|_2 \leq \infty, \vec{x}(0)=0}}{\operatorname{argmin}} \underset{\vec{d}}{\operatorname{argmax}} \frac{\|\vec{e}\|_2^2}{\|\vec{d}\|_2^2} \quad (4.163)$$

which can be a difficult problem to solve for general $F_L(G, K)$.

Thus, one can restate the problem by noting that, one has for all \vec{d} with $\|\vec{d}\|_2 < \infty$

$$\|F_L(G, K)\|_\infty \geq \frac{\|\vec{e}\|_2}{\|\vec{d}\|_2} \quad (4.164)$$

Thus, one method to solve the \mathcal{H}_∞ OCP is to bound \mathcal{H}_∞ -norm by some $\gamma > 0$, i.e.

$$\|F_L(G, K)\|_\infty^2 \leq \gamma^2 \quad (4.165)$$

Then, one has

$$\gamma^2 \|\vec{d}\|_2^2 \geq \|\vec{e}\|_2^2 \quad (4.166)$$

$$\|\vec{e}\|_2^2 - \gamma^2 \|\vec{d}\|_2^2 \leq 0 \quad (4.167)$$

which will equal zero for some worst case \vec{d} and $\gamma = \gamma_{min} = \|F_L(G, K)\|_\infty$.

Thus, the \mathcal{H}_∞ OCP can be reformulated as solving for the optimal controller *and* maximizing disturbance that solves the following constrained min-max OCP.

$$\begin{aligned}K^{opt}, \vec{d}^{opt} &= \underset{\substack{K \text{ stabilizing} \\ \vec{d}}}{\operatorname{argmin}} \underset{\vec{d}}{\operatorname{argmax}} \mathcal{J}(K, \vec{d}) \\ \text{subject to : } \gamma &\geq \|F_L(G, K)\|_\infty\end{aligned}\quad (4.168)$$

with cost functional

$$\mathcal{J}(K, \vec{d}) = \frac{1}{2} \int_0^{t_f} \vec{e}^T \vec{e} - \gamma^2 \vec{d}^T \vec{d} d\tau \quad (4.169)$$

where notably $\vec{x}_0 = 0$, t_f is given and may be finite in this reformulation, and $\vec{x}(t_f)$ is free to vary in the optimization, though one can include a terminal cost on $\vec{x}(t_f)$ if desired.

If one assumes $C_2 = I$, $D_{21} = D_{22} = 0$, for state feedback \mathcal{H}_∞ optimal control, $\vec{u}(t) = D_K(t)\vec{x}(t)$, then this cost functional becomes

$$\mathcal{J}(\vec{u}, \vec{d}) = \frac{1}{2} \int_0^{t_f} [C_1 \vec{x} + D_{12} \vec{u} + D_{11} \vec{d}]^T [C_1 \vec{x} + D_{12} \vec{u} + D_{11} \vec{d}] - \gamma^2 \vec{d}^T \vec{d} d\tau \quad (4.170)$$

$$\begin{aligned}\mathcal{J}(\vec{u}, \vec{d}) &= \frac{1}{2} \int_0^{t_f} \vec{x}^T C_1^T C_1 \vec{x} + 2 \vec{x}^T [C_1^T D_{12} \quad C_1^T D_{11}] \begin{bmatrix} \vec{u} \\ \vec{d} \end{bmatrix} \\ &\quad + \begin{bmatrix} \vec{u} \\ \vec{d} \end{bmatrix}^T [D_{12}^T D_{12} \quad D_{12}^T D_{11} \quad D_{11}^T D_{11} - \gamma^2 I] \begin{bmatrix} \vec{u} \\ \vec{d} \end{bmatrix} d\tau\end{aligned}\quad (4.171)$$

which is close to a quadratic cost functional except for the γ^2 term which is to be minimized as part of the optimization, and thus is not truly a quadratic cost.

ARE Synthesis for State Feedback \mathcal{H}_∞ OCP

However, assuming one has a constant upper bound γ , then one has the sub-problem to find the \vec{d} and \vec{u} which maximize and minimize the cost functional to obtain $D_K(t)$. In this way, one can define the LQR sub-problem with

$$Q = C_1^T C_1 \quad (4.172)$$

$$S = [C_1^T D_{12} \quad C_1^T D_{11}] \quad (4.173)$$

$$R = \begin{bmatrix} D_{11}^T D_{12} & D_{11}^T d_2 \\ D_{11}^T D_{12} & D_{11}^T D_{11} - \gamma^2 I \end{bmatrix} \quad (4.174)$$

$$\tilde{B} = [B_2 \quad B_1] \quad (4.175)$$

and

$$\tilde{u} = \begin{bmatrix} \vec{u} \\ \vec{d} \end{bmatrix} \quad (4.176)$$

Then, one has the LQR cost functional

$$\mathcal{J}(\tilde{u}) = \frac{1}{2} \int_0^{t_f} \vec{x}^T Q \vec{x} + 2 \vec{x}^T S \tilde{u} + \tilde{u}^T R \tilde{u} d\tau \quad (4.177)$$

which is solved by the differential Riccati equation

$$-\dot{P} = PA + A^T P + Q - (P\tilde{B} + S)R^{-1}(\tilde{B}^T P + S^T) \quad (4.178)$$

with the optimal \tilde{u}^{opt} given by

$$\tilde{u} = -R^{-1}(\tilde{B}^T P(t) + S^T) \vec{x} \quad (4.179)$$

and the optimal control sequence \vec{u} is

$$\begin{aligned} \vec{u} &= [I_{n_u \times n_u} \quad 0] \tilde{u} \\ &= -[I_{n_u \times n_u} \quad 0] R^{-1}(\tilde{B}^T P(t) + S^T) \vec{x} \\ &= D_K(t) \vec{x} \end{aligned} \quad (4.180)$$

For $t_f \rightarrow \infty$, D_K is a fixed-gain controller obtained by the Riccati matrix $P \geq 0$ that solves the CARE

$$0 = PA + A^T P + Q - (P\tilde{B} + S)R^{-1}(\tilde{B}^T P + S^T) \quad (4.181)$$

which can be rewritten as

$$PA + A^T P + C_1^T C_1 - \begin{bmatrix} B_2^T P + D_{12}^T C_1 \\ B_1^T P + D_{11}^T C_1 \end{bmatrix}^T \begin{bmatrix} D_{12}^T D_{12} & D_{12}^T D_{11} \\ D_{11}^T D_{12} & D_{11}^T D_{11} - \gamma^2 I \end{bmatrix} \begin{bmatrix} B_2^T P + D_{12}^T C_1 \\ B_1^T P + D_{11}^T C_1 \end{bmatrix} = 0 \quad (4.182)$$

Note that for this LQR sub-problem to be well-posed, one must assume that for the LTI plant state-space model (A, B_2, C_1, D_{12}) there are no zeros on the $j\omega$ axis, (A, B_2) is stabilizable, (A, C_1) is detectable, and $(D_{12}^T D_{12})^{-1}$ exists (i.e. injective). However, choosing a γ to solve the \mathcal{H}_∞ OCP is still required for this LQR solution method and is typically done through **γ -iteration**

A simple iteration procedure can be performed as a bisection search as

1. Initialize γ larger than the anticipated optimal γ for binary search
 - Form LQR cost matrices using γ
 - Solve continuous-time algebraic Riccati equation (CARE) for matrix P
 - If $P > 0$ and $(A - BD_K)$ Hurwitz:
 - Decrease γ by bisection (until convergence threshold)
 - Else:
 - Increase γ by bisection
2. Convergence to γ_{min} to form D_K

Note that care must be taken as γ approaches γ_{min} as R typically becomes ill-conditioned. It is also typically prudent to slightly increase γ from γ_{min} to reduce feedback gain magnitudes and improve the accuracy of the numerical solver for the CARE.

SDP Synthesis for State Feedback \mathcal{H}_∞ OCP

The Bounded Real Lemma states that A_L is Hurwitz, and $\|F_L(G, K)\|_\infty < \gamma^2$ if and only if there exists $P > 0$ satisfying the strict ARI:

$$A_L^T P + P A_L + C_L^T C_L + \gamma^{-2} P B_L B_L^T P < 0 \quad (4.183)$$

which can be written as

$$\begin{bmatrix} (A + B_2 D_K)^T P + P(A + B_2 D_K) & PB_1 \\ B_1^T P & \gamma^{-2} I \end{bmatrix} + \begin{bmatrix} (C_1 + D_{12} D_K)^T \\ 0 \end{bmatrix} \begin{bmatrix} C_1 + D_{12} D_K & 0 \end{bmatrix} < 0 \quad (4.184)$$

which is a nonlinear matrix inequality in (P, D_K) due to the bilinear and quadratic terms. However, one can use the **symmetric congruence transformation**

$$\begin{bmatrix} P^{-1} & 0 \\ 0 & I \end{bmatrix} \quad (4.185)$$

which can be left and right multiplied to get

$$\begin{bmatrix} P^{-1}(A + B_2 D_K)^T + (A + B_2 D_K)P^{-1} & B_1 \\ B_1^T & \gamma^{-2} I \end{bmatrix} + \begin{bmatrix} P^{-1}(C_1 + D_{12} D_K)^T \\ 0 \end{bmatrix} \begin{bmatrix} (C_1 + D_{12} D_K)P^{-1} & 0 \end{bmatrix} < 0 \quad (4.186)$$

Next, defining a change of variables as $Q = P^{-1}$ and $R = D_K Q$, one has alternatively

$$\begin{bmatrix} QA^T + AQ + R^T B_2^T + B_2 R & B_1 \\ B_1^T & \gamma^{-2} I \end{bmatrix} + \begin{bmatrix} (C_1 Q + D_{12} R)^T \\ 0 \end{bmatrix} \begin{bmatrix} C_1 Q + D_{12} R & 0 \end{bmatrix} < 0 \quad (4.187)$$

which is still not an LMI due to the quadratic term, $(C_1 Q + D_{12} R)^T (C_1 Q + D_{12} R)$. However, by the Schur Complement Lemma, one can write

$$\begin{bmatrix} QA^T + AQ + R^T B_2^T + B_2 R & B_1 & (C_1 Q + D_{12} R)^T \\ B_1^T & \gamma^{-2} I & 0 \\ C_1 Q + D_{12} R & 0 & -I \end{bmatrix} < 0 \quad (4.188)$$

Thus, one can state the \mathcal{H}_∞ OCP as an SDP in (Q, R, γ) , i.e.

$$(Q, R, \gamma)^{opt} = \underset{\substack{Q \in \mathbb{S}^{n_x}, R \in \mathbb{R}^{n_u \times n_x}, \gamma > 0}}{\operatorname{argmin}} \gamma$$

subject to:
$$\begin{bmatrix} QA^T + AQ + R^T B_2^T + B_2 R & B_1 & (C_1 Q + D_{12} R)^T \\ B_1^T & -\gamma^{-2} I & 0 \\ C_1 Q + D_{12} R & 0 & -I \end{bmatrix} < 0$$

$$-Q < 0$$
(4.189)

with the state feedback \mathcal{H}_∞ optimal controller, K^{opt} , as $\vec{u}(t) = D_K \vec{x}(t)$ reconstructed with

$$D_K = RQ^{-1} \quad (4.190)$$

Signal-Weighted \mathcal{H}_∞ Optimal Control Design

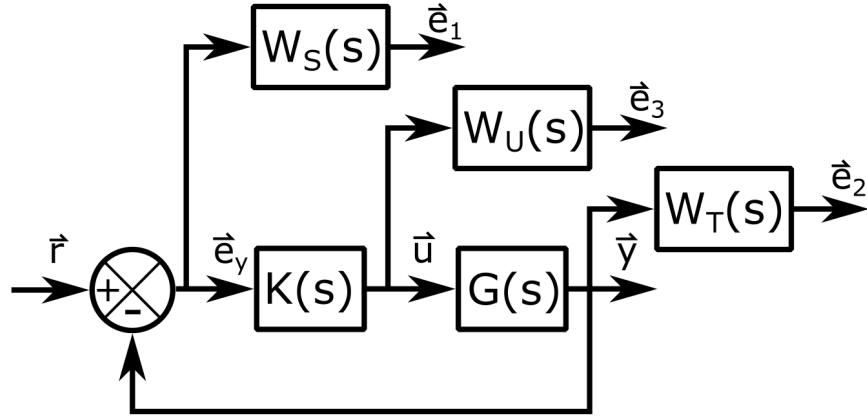
Recall in classical control, one can design the feedback control policy, $K(s)$, by shaping the frequency response of the loop gain, $|L(j\omega)|$, to meet robust control system performance and robust stability requirements, i.e. loop-shaping. Note that in SISO loop-shaping, $L(s) = G(s)K(s)$ is the open-loop transfer function with singular value $\sigma(L) = \bar{\sigma}(L) = |L|$ that can be related to the error/sensitivity transfer function, i.e. $S(s) = (1 + L(s))^{-1}$, and the closed-loop/complementary sensitivity transfer function, $T(s) = L(s)(1+L(s))^{-1}$, where $S(s) + T(s) = 1$ must hold. Here, one shapes the loop gain through selecting the *control stages* of $K(s)$ to set the crossover frequency, increase the gain at low frequencies, decrease the gain at high frequencies, and reduce the slope of the crossover region to achieve good classical stability margins. Furthermore, to check the control effort requires one to also assess that $K(s)S(s)$ separately from the loop-shaping procedure.

This frequency-domain robust control design can be generalized to MIMO LTI systems through \mathcal{H}_∞ optimal control and to meet robust MIMO system performance and robust stability requirements. For MIMO LTI systems, instead of using separate control stage transfer functions for the controller, one can use different *weight transfer functions* on the different signals which make up the both the generalized disturbances, \vec{d} , and generalized errors, \vec{e} , for the generalized LTI feedback control system $F_L(G, K)$ so that $F_L(G, K)$ is stable and

$$\left\| \frac{\vec{e}}{\vec{d}} \right\|_\infty = \|F_L(G, K)\|_\infty \leq \gamma \quad (4.191)$$

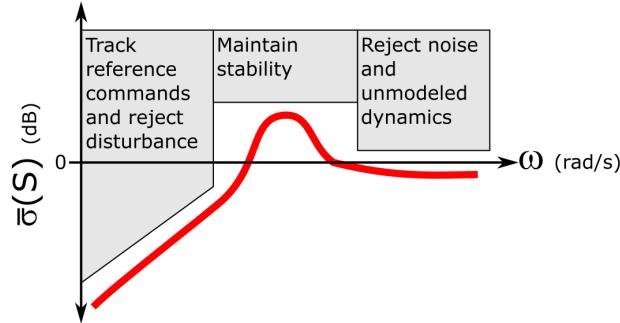
In general, these weight transfer functions are selected as functions of frequency which allow one to normalize and balance the various robust stability and performance requirements across the expected frequency ranges, e.g. reference commands, plant disturbances, control effort, error tracking, model-following, state regulation. In some cases, one can choose the weight transfer functions such that if $\gamma = 1$, then all robust stability and performance requirements are met. Thus, \mathcal{H}_∞ optimal control design is quite varied and is typically problem-specific.

However, an introductory method for a **single degree-of-freedom LTI feedback control system** is **mixed-sensitivity \mathcal{H}_∞ loop-shaping** which forms the following signal-weighted LTI feedback control system:

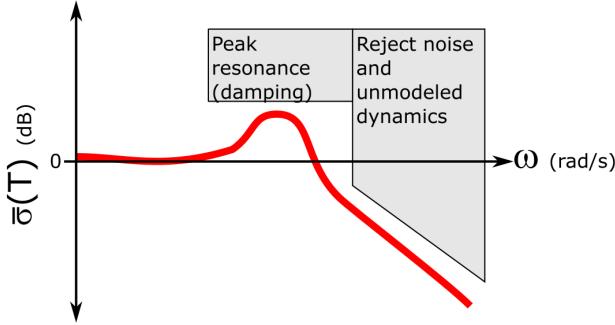


where one chooses the weights, W_S , W_u , and W_T on the signals, \vec{e}_y , $u(s)$, and \vec{y} , respectively, to form the generalized error vector, \vec{e} as three vectors, \vec{e}_1 , \vec{e}_2 , and \vec{e}_3 . Note that \vec{e}_y is the tracking error and \vec{r} is the only generalized disturbance.

The error or sensitivity transfer function matrix is $S(s) = (I + K(s)G(s))^{-1}$ and should generally follow the shape



and the closed-loop or complementary sensitivity matrix is $T(s) = (I + K(s)G(s))^{-1}L(s)$ and should generally follow the shape



At low frequency, $S(s)$ needs to be small as $S(s)$ is the transfer function for the tracking error and $T(s)$ is near unity or 0 dB as $S(s) + T(s) = 1$. At high frequencies, $T(s)$ needs to roll off to reject high-frequency noise and unmodeled dynamics and $S(s)$ is near unity or 0 dB as $S(s) + T(s) = 1$. Furthermore, the near-singularity of $S^{-1}(s) = I + K(s)G(s)$ is measured by the minimum of $\underline{\sigma}(I + K(j\omega)G(j\omega))$ across all frequencies which equates to the maximum or peak value of $S(j\omega)$, i.e. $\|S\|_\infty$. At the peak of $T(j\omega)$, $\|T\|_\infty$ denotes the closed-loop system's peak resonance at the frequency for which inputs are most amplified. Thus $\|T\|_\infty$ is analogous to the idea of damping for second-order systems, i.e. the dominant poles are close to the $j\omega$ -axis at this frequency.

With this frequency-dependent weights framework, the \mathcal{H}_∞ optimal control synthesis will provide γ_{min} such that to minimize

$$\begin{aligned} \|W_S S\|_\infty &\leq \gamma \\ \|W_u K S\|_\infty &\leq \gamma \\ \|W_T T\|_\infty &\leq \gamma \end{aligned} \tag{4.192}$$

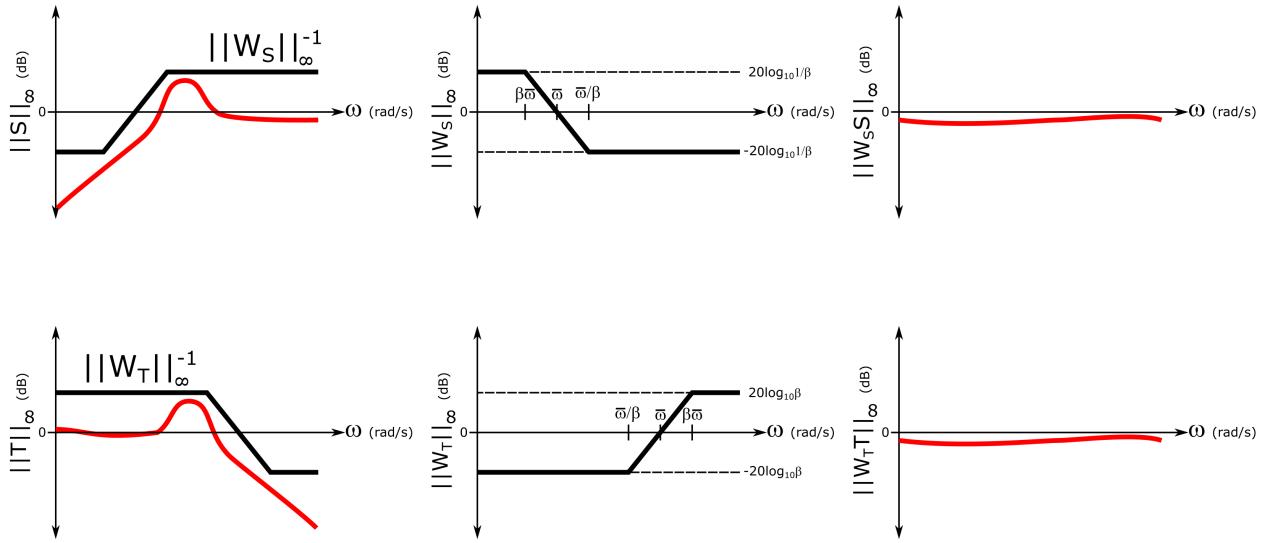
which equates to

$$\begin{aligned} \|S\|_\infty &\leq \gamma |W_S^{-1}| \\ \|K S\|_\infty &\leq \gamma |W_u^{-1}| \\ \|T\|_\infty &\leq \gamma |W_T^{-1}| \end{aligned} \tag{4.193}$$

where one typically desires to minimize the control effort, $\|u\|$, across all frequencies to ensure that any actuators are not position or rate saturated. Thus, one typically sets a constant weight on W_u . Here the selection of the W_S and W_T weighting filters are first-order lag and lead transfer functions, i.e.

$$W(s) = \frac{\beta s + \bar{\omega}}{s + \beta \bar{\omega}} \tag{4.194}$$

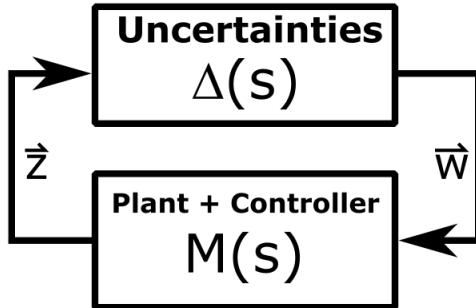
where for $0 < \beta < 1$, this corresponds to a lag control (e.g. W_S) and $\beta > 1$ corresponds to a lead control (e.g. W_T). This corresponds to the following diagrams:



The degree of difficulty in choosing these weighting filters can be comparable to choosing the proportional gain, low-frequency boost, and lead control stages in SISO loop-shaping design. Similarly, design iteration with the weighting filters is used in conjunction with the \mathcal{H}_∞ optimal control synthesis to perform mixed-sensitivity \mathcal{H}_∞ loop-shaping control design.

μ -Synthesis via D-K Iteration

To incorporate robust stability margins directly in the optimal control framework, recall the generalized ΔM framework as



where the structured singular value (SSV) μ_Δ can be used to compute an upper and lower bound for the inverse of the minimum possible $\bar{\sigma}\Delta$ which causes the ΔM model to become unstable. This is related to the maximum singular value of M by

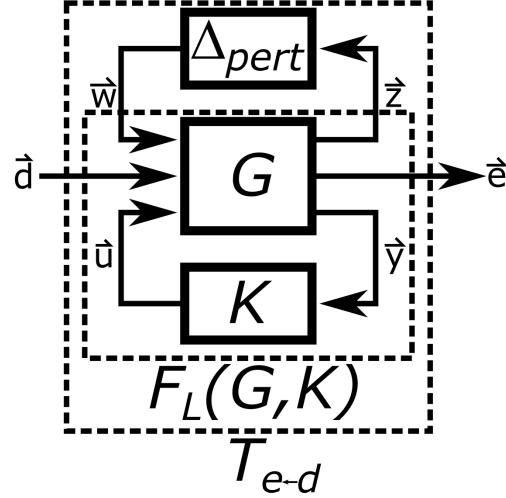
$$\bar{\rho}(M) \leq \mu_\Delta(M) \leq \bar{\sigma}(M) \quad (4.195)$$

which is numerically approximated by

$$\max_Q \lambda(QM) \leq \mu_\Delta(M) \leq \inf_D \bar{\sigma}(DMD^{-1}) \quad (4.196)$$

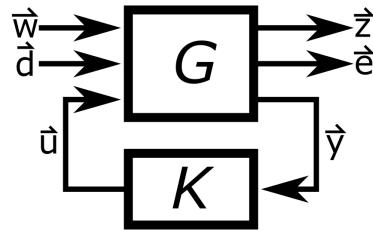
where the frequency-dependent D matrices, which commute with Δ (i.e. $D\Delta = \Delta D$), are called the **D scalings**.

This ΔM framework can be combined with the \mathcal{H}_∞ optimal control framework as the general block diagram



where \vec{d} and \vec{e} are the generalized disturbance and generalized error, \vec{w} and \vec{z} are the perturbation input and output vectors to some plant perturbation uncertainty, Δ_{pert} , and \vec{u} and \vec{y} are the control input to the plant and plant output to the controller vectors as previously. Note that the lower LFT, $F_L(G, K)$, is the equivalent to M in the ΔM robust analysis model.

Here, one can then set up the model $F_L(G, K)$ as



where one has for the augmented uncertainty block matrix, Δ , in the ΔM robust analysis

$$\Delta = \begin{bmatrix} \Delta_{pert} & 0 \\ 0 & \Delta_F \end{bmatrix} \quad (4.197)$$

which takes into account both input and output vectors in the μ -synthesis model as $\vec{e} = \Delta_F \vec{d}$. Thus, the μ -**synthesis** OCP attempts to minimize over all stabilizing controllers, K , the peak value of μ_Δ of the closed-loop transfer function, i.e.

$$K^{opt} = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \max_{\omega} \mu_\Delta (F_L(G, K)(j\omega)) \quad (4.198)$$

To derive a tractable approximation for this μ -synthesis optimization known as $D - K$ iteration, recall the upper bound definition for μ_Δ , one can rewrite this minimization of the maximum possible μ_Δ as

$$K^{opt} = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \max_{\omega} \min_{D_\omega} \bar{\sigma} \left(D_\omega F_L(G, K)(j\omega) D_\omega^{-1} \right) \quad (4.199)$$

where D_ω is chosen from all possible scalings independently as every ω . Then, rearranging, one has

$$K^{opt} = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \min_{D_\omega} \max_{\omega} \bar{\sigma} \left(D_\omega F_L(G, K)(j\omega) D_\omega^{-1} \right) \quad (4.200)$$

or

$$K^{opt} = \underset{K \text{ stabilizing}}{\operatorname{argmin}} \min_{D_\omega} \|D_\omega F_L(G, K) D_\omega^{-1}\|_\infty \quad (4.201)$$

Thus, one can see that this optimization can be constructed as minimizing two different parameters, D and K , which is done as an iterative procedure, i.e. holding D as fixed and finding the optimal K using \mathcal{H}_∞ , then holding K fixed and finding the optimal D that minimizes the transformed \mathcal{H}_∞ -norm. Note that this D procedure serves as an approximation for maximizing the μ_Δ upper bound, i.e. maximizing the “distance” to singularity/instability of $F_L(G, K)$ for the unstructured uncertainty Δ (i.e. fully complex block matrix as defined previously). This approximation is typically very close. However, it should also be noted that $D - K$ iteration is not guaranteed to converge to a global or even a local minimum which is a serious shortcoming of this approach, but has been shown to work well in many different flight vehicle control problems including highly flexible airplanes, missile autopilots, modern fighter airplanes, and the space shuttle flight control system.

4.6 Robust Optimal Control of LPV Systems

A special class of LTV systems are **linear, parameter-varying (LPV)** systems, represented by the following linear state-space model

$$\begin{aligned} \dot{\vec{x}}(t) &= A(\vec{\beta}(t)) \vec{x}(t) + B(\vec{\beta}(t)) \vec{u}(t) \\ \vec{y}(t) &= C(\vec{\beta}(t)) \vec{x}(t) + D(\vec{\beta}(t)) \vec{u}(t) \end{aligned} \quad (4.202)$$

where $\vec{\beta}(t)$ is the **parameter vector** that must be a continuously differentiable function of time and the state-space matrices are also continuous functions of $\vec{\beta}$. Both of these a LPV systems typically arise when one linearizes a nonlinear system about equilibrium points specified by certain parameters, i.e. $\vec{\beta}$.

This is typically at least restricted by some upper and lower bounds, e.g. $\vec{\beta}_U$ and $\vec{\beta}_L$, respectively, and may also be rate restricted by some upper and lower bounds, e.g. $\dot{\vec{\beta}}_U$ and $\dot{\vec{\beta}}_L$. Thus, one can write the set

of admissible trajectories for $\vec{\beta}$, \mathcal{B} , as

$$\mathcal{B} = \left\{ \vec{\beta}(t) : \mathbb{R}_+ \rightarrow \mathbb{R}^{n_\beta} \text{ such that: } \vec{\beta}(t) \text{ continuously differentiable,} \right. \\ \left. \vec{\beta}_L \leq \vec{\beta}(t) \leq \vec{\beta}_U \forall t \geq 0, \quad \dot{\vec{\beta}}_L \leq \dot{\vec{\beta}}(t) \leq \dot{\vec{\beta}}_U \forall t \geq 0 \right\} \quad (4.203)$$

Robust LPV System Control Design

However, more advanced approaches for LPV systems have been proposed based on the specific structure of the state-space matrices, e.g. linear or rational dependence on $\vec{\beta}$. For arbitrary dependence, one can generally specify the performance of an LPV system, G_β in terms of its maximum possible induced $\mathcal{L}_{2 \leftarrow 2}$ gain from input \vec{u} to output \vec{y} , i.e.

$$\|G_\beta\|_{2 \leftarrow 2} = \max_{\vec{\beta} \in \mathcal{B}, 0 \neq \|\vec{u}\|_2 \leq \infty, \vec{x}(0)=0} \frac{\|\vec{y}\|_2}{\|\vec{u}\|_2} \quad (4.204)$$

As G_β is time-varying, one *cannot* interpret this as an \mathcal{H}_∞ -norm in the frequency domain. However, one can use a **generalized Bounded Real Lemma** to derive sufficient conditions for an upper bound on $\|G_\beta\|_{2 \leftarrow 2}$.

In particular, a rate-unbounded LPV system is globally uniformly asymptotically stable and $\|G_\beta\|_{2 \leftarrow 2} \leq \gamma$ if there exists $P > 0$ such that $\forall \vec{\beta} \in [\vec{\beta}_L, \vec{\beta}_U]$, one has

$$\begin{bmatrix} A^T(\vec{\beta})P + PA(\vec{\beta}) & PB(\vec{\beta}) \\ B^T(\vec{\beta})P & -I \end{bmatrix} + \frac{1}{\gamma^2} \begin{bmatrix} C^T(\vec{\beta}) \\ D^T(\vec{\beta}) \end{bmatrix} [C(\vec{\beta}) \quad D(\vec{\beta})] < 0 \quad (4.205)$$

where the exponential stability notion by noting the definition is held by the upper left block of the first term. To also show that $\|G_\beta\|_{2 \leftarrow 2} \leq \gamma$ note that one can left and right multiply by the stacked state and input vectors, $[\vec{x}^T \quad \vec{u}^T]^T$, to obtain

$$\dot{\vec{x}}^T P \vec{x} + \vec{x}^T P \dot{\vec{x}} + \frac{1}{\gamma^2} \vec{y}^T \vec{y} - \vec{u}^T \vec{u} \leq 0 \quad (4.206)$$

$$\dot{V} + \frac{1}{\gamma^2} \vec{y}^T \vec{y} - \vec{u}^T \vec{u} \leq 0 \quad (4.207)$$

and integrating from $t=0$ to $t=t_f$, one has

$$V(\vec{x}(t_f)) - V(\vec{x}(0)) + \frac{1}{\gamma^2} \int_0^{t_f} \vec{y}^T \vec{y} dt - \int_0^{t_f} \vec{u}^T \vec{u} dt \leq 0 \quad (4.208)$$

where $V(\vec{x}(t_f)) > 0$ and $V(\vec{x}(0)) = 0$, thus

$$\int_0^{t_f} \vec{y}^T \vec{y} dt \leq \gamma^2 \int_0^{t_f} \vec{u}^T \vec{u} dt \quad (4.209)$$

Then, as $t_f \rightarrow \infty$, one has

$$\|\vec{y}\|_2^2 \leq \gamma^2 \|\vec{u}\|_2^2 \quad (4.210)$$

$$\|G_\beta\|_{2 \leftarrow 2} \leq \gamma \quad (4.211)$$

The matrix inequality is a parameterized LMI condition, i.e. one LMI for each value of $\vec{\beta} \in \mathcal{B}$. In practice, this infinite collection of LMI conditions is approximated by enforcing them only on a finite grid of points. Then, the finite dimensional LMI conditions can be directly obtained to bound $\|G_\beta\|_{2 \leftarrow 2}$ without approximation if the state matrices have a rational dependence on $\vec{\beta}$.

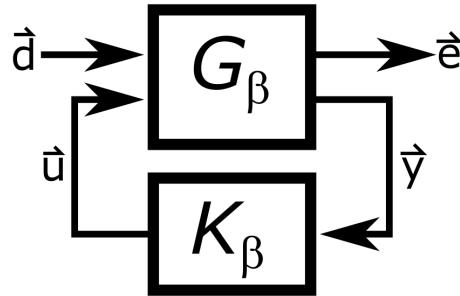
For rate-bounded LPV system, one can alter the matrix inequality with a slight variation on the Lyapunov theory, which requires one to check $2^{n\beta}$ LMI conditions evaluated at the endpoints defined by the hypercube of the elements of $\vec{\beta}_U$ and $\vec{\beta}_L$. Furthermore, in this case, one also needs to search over the infinitely dimensional space of functions, $P(\beta)$, which must be restricted to a finite dimensional subspace. In practice, one specifies a collection of scalar basis functions, $g_i(\vec{\beta})$, and use a linear combination of these basis functions

$$P(\vec{\beta}) = \sum_{i=1}^N g_i(\vec{\beta}) P_i \quad (4.212)$$

where P_i are symmetric matrices that form the finite collection of decision variables. Often polynomials are chosen as basis functions.

SDP Solution for Optimal State Feedback LPV Control

Consider the following LPV feedback control system



with LPV plant

$$\begin{aligned} \dot{\vec{x}}(t) &= A(\vec{\beta})\vec{x}(t) + [B_1(\vec{\beta}) \quad B_2(\vec{\beta})] \begin{bmatrix} \vec{d}(t) \\ \vec{u}(t) \end{bmatrix} \\ \begin{bmatrix} \vec{e}(t) \\ \vec{y}(t) \end{bmatrix} &= \begin{bmatrix} C_1(\vec{\beta}) \\ I \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 & D_{12}(\vec{\beta}) \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \vec{d}(t) \\ \vec{u}(t) \end{bmatrix} \end{aligned} \quad (4.213)$$

While optimal LPV controllers can also be designed for output feedback and for observer feedback, consider the state feedback LPV controller

$$\vec{u} = D_K(\vec{\beta})\vec{x} \quad (4.214)$$

This results in a closed-loop LPV system

$$\begin{aligned}\dot{\vec{x}}(t) &= \left(A(\vec{\beta}) + B_2(\vec{\beta})D_K(\vec{\beta}) \right) \vec{x}(t) + B_1(\vec{\beta})\vec{d}(t) \\ \vec{e}(t) &= \left(C_1(\vec{\beta}) + D_{12}(\vec{\beta})D_K(\vec{\beta}) \right) \vec{x}(t)\end{aligned}\quad (4.215)$$

Then, by the generalized bounded real lemma, the closed-loop LPV system is internally stable and $\|G_\beta\|_{2 \leftarrow 2} \leq \gamma$ if there exists $P > 0$ such that $\forall \vec{\beta} \in [\vec{\beta}_L, \vec{\beta}_U]$, one has

$$\begin{bmatrix} A_L^T(\vec{\beta})P + PA_L(\vec{\beta}) & PB_L(\vec{\beta}) \\ B_L^T(\vec{\beta})P & -I \end{bmatrix} + \frac{1}{\gamma^2} \begin{bmatrix} C_L^T(\vec{\beta}) \\ D_L^T(\vec{\beta}) \end{bmatrix} [C_L(\vec{\beta}) \quad D_L(\vec{\beta})] < 0 \quad (4.216)$$

which can be written as the following using a change of variables

$$Q = P^{-1} \quad (4.217)$$

and

$$R(\vec{\beta}) = D_K(\vec{\beta})Q \quad (4.218)$$

and the Schur Complement Lemma to get

$$\begin{bmatrix} QA^T(\vec{\beta}) + A(\vec{\beta})Q + R^T(\vec{\beta})B_2^T(\vec{\beta}) + B_2(\vec{\beta})R(\vec{\beta}) & B_1(\vec{\beta}) & (C_1(\vec{\beta})Q + D_{12}(\vec{\beta})R(\vec{\beta}))^T \\ B_1^T(\vec{\beta}) & \gamma^{-2}I & 0 \\ C_1(\vec{\beta})Q + D_{12}(\vec{\beta})R(\vec{\beta}) & 0 & -I \end{bmatrix} < 0 \quad (4.219)$$

Thus, one can state the state feedback LPV OCP as an SDP in (Q, R, γ) , i.e.

$$\begin{aligned}(Q, R(\vec{\beta}), \gamma)^{opt} &= \underset{Q \in \mathbb{S}^{n_x}, R \in \mathbb{R}^{n_u \times n_x}, \gamma > 0}{\operatorname{argmin}} \gamma \\ \text{subject to: } &\begin{bmatrix} QA^T(\vec{\beta}) + A(\vec{\beta})Q + R^T(\vec{\beta})B_2^T(\vec{\beta}) + B_2(\vec{\beta})R(\vec{\beta}) & B_1(\vec{\beta}) & (C_1(\vec{\beta})Q + D_{12}(\vec{\beta})R(\vec{\beta}))^T \\ B_1^T(\vec{\beta}) & -\gamma^{-2}I & 0 \\ C_1(\vec{\beta})Q + D_{12}(\vec{\beta})R(\vec{\beta}) & 0 & -I \end{bmatrix} < 0 \\ Q &> 0\end{aligned}\quad (4.220)$$

with the state feedback LPV optimal controller as $\vec{u}(t) = D_K(\vec{\beta})\vec{x}(t)$ reconstructed with

$$D_K(\vec{\beta}) = R(\vec{\beta})Q^{-1} \quad (4.221)$$

which must be approximately solved by gridding over $[\vec{\beta}_L, \vec{\beta}_U]$.

Discrete-Time Systems and Linear Control Theory

5.1 Discrete-Time Linear Systems

A multivariate extension of difference equations or recursive relations for discrete-time MIMO systems is the **discrete-time state-space model** which can be defined as the general form of two equations

$$\begin{aligned}\vec{x}[k+1] &= f(k, \vec{x}[k], \vec{u}[k]) \\ \vec{y} &= h(k, \vec{x}, \vec{u})\end{aligned}\tag{5.1}$$

where $k \in \mathbb{Z}$ is the integer-valued **time step**, $f : \mathbb{Z} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_x}$, $h : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_y}$, $\vec{u} \in \mathbb{R}^{n_u}$ is the **input vector** of n_u input signals, $\vec{y} \in \mathbb{R}^{n_y}$ is the **output vector** of n_y output signals, and $\vec{x} \in \mathbb{R}^{n_x}$ is the **state vector** of the n_x^{th} -order dynamical system. Just as for continuous-time systems, the first vector-valued difference equation is the **discrete-time dynamics equation**, also known as the **discrete-time state equation**, and the second vector-valued algebraic equation is the **output equation**. Contrary to continuous-time models, the recursive nature of discrete-time allows one to easily solve for $\vec{x}[k]$ given any $\vec{x}[0]$.

The **discrete-time time-invariant (TI) state-space models** have the general form

$$\begin{aligned}\vec{x}[k+1] &= f(\vec{x}[k], \vec{u}[k]) \\ \vec{y}[k] &= h(\vec{x}[k], \vec{u}[k])\end{aligned}\tag{5.2}$$

The **discrete-time linear state-space models** have the general form

$$\begin{aligned}\dot{\vec{x}}[k+1] &= F[k]\vec{x}[k] + G[k]\vec{u}[k] \\ \vec{y}[k] &= H[k]\vec{x}[k] + D[k]\vec{u}[k]\end{aligned}\tag{5.3}$$

where $A \in \mathbb{R}^{n_x \times n_x}$ matrix is the **discrete-time state matrix**. The $G \in \mathbb{R}^{n_x \times n_u}$ matrix is the **discrete-time input matrix**. The $H \in \mathbb{R}^{n_y \times n_x}$ matrix is the **discrete-time output matrix** and $D \in \mathbb{R}^{n_y \times n_u}$ is the **discrete-time feedthrough matrix**.

The **discrete-time linear time-invariant (LTI) state-space models** have the general form

$$\begin{aligned}\vec{x}[k+1] &= F\vec{x}[k] + G\vec{u}[k] \\ \vec{y}[k] &= H\vec{x}[k] + D\vec{u}[k]\end{aligned}\quad (5.4)$$

Discrete-Time Trim and Linearization

For discrete-time state-space time-invariant systems similar to continuous-time time-invariant systems, a state-input pair $(\bar{\vec{x}}, \bar{\vec{u}})$ is an **equilibrium point** if $\vec{x}_k = \bar{\vec{x}}$ is zero for all $k \geq 0$, i.e. if

$$f(k, \bar{\vec{x}}, \bar{\vec{u}}) = \bar{\vec{x}} \quad (5.5)$$

is a valid solution for all $k \geq 0$. Similarly, if one initializes $\vec{x}[k] = \bar{\vec{x}}$ at $k = 0$ and sets $\vec{u}[k] = \bar{\vec{u}}$ for $k \geq 0$, then $\vec{x}[k] = \bar{\vec{x}}$ and $\vec{y} = h(\bar{\vec{x}}, \bar{\vec{u}})$ for all $k \geq 0$, i.e., \vec{x} is “steady.”

Defining the state, input, and output **perturbation vectors** about constants $\bar{\vec{x}}$, $\bar{\vec{u}}$, and $\bar{\vec{y}}$ as

$$\Delta\vec{x}[k] = \vec{x}[k] - \bar{\vec{x}} \quad (5.6)$$

$$\Delta\vec{u}[k] = \vec{u}[k] - \bar{\vec{u}} \quad (5.7)$$

$$\Delta\vec{y}[k] = \vec{y}[k] - \bar{\vec{y}} \quad (5.8)$$

and recognizing for trim, $f(\bar{\vec{x}}, \bar{\vec{u}}) = \bar{\vec{x}}$ and $h(\bar{\vec{x}}, \bar{\vec{u}}) = \bar{\vec{y}}$, one has

$$\Delta\vec{x}[k+1] = \vec{x}[k+1] - \bar{\vec{x}} = f(\vec{x}, \vec{u}) - f(\bar{\vec{x}}, \bar{\vec{u}}) = \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta\vec{x}[k] + \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta\vec{u}[k] + \text{HOT} \quad (5.9)$$

$$\Delta\vec{y}[k] = \vec{y}[k] - \bar{\vec{y}} = h(\vec{x}, \vec{u}) - h(\bar{\vec{x}}, \bar{\vec{u}}) = \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta\vec{x}[k] + \left[\frac{\partial h}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] \Delta\vec{u}[k] + \text{HOT} \quad (5.10)$$

by the Taylor series expansion.

Next, setting

$$F = \left[\frac{\partial f}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_1}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_x}}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_{n_x}}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \quad (5.11)$$

$$G = \left[\frac{\partial f}{\partial \vec{u}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] = \begin{bmatrix} \frac{\partial f_1}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_1}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial f_{n_u}}{\partial u_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial f_{n_u}}{\partial u_{n_u}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \quad (5.12)$$

$$H = \left[\frac{\partial h}{\partial \vec{x}}(\bar{\vec{x}}, \bar{\vec{u}}) \right] = \begin{bmatrix} \frac{\partial h_1}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_1}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial x_1}(\bar{\vec{x}}, \bar{\vec{u}}) & \cdots & \frac{\partial h_{n_y}}{\partial x_{n_x}}(\bar{\vec{x}}, \bar{\vec{u}}) \end{bmatrix} \quad (5.13)$$

$$D = \left[\frac{\partial h}{\partial \vec{u}}(\vec{x}, \vec{u}) \right] = \begin{bmatrix} \frac{\partial h_1}{\partial u_1}(\vec{x}, \vec{u}) & \cdots & \frac{\partial h_1}{\partial u_{n_u}}(\vec{x}, \vec{u}) \\ \vdots & \ddots & \vdots \\ \frac{\partial h_{n_y}}{\partial u_1}(\vec{x}, \vec{u}) & \cdots & \frac{\partial h_{n_y}}{\partial u_{n_u}}(\vec{x}, \vec{u}) \end{bmatrix} \quad (5.14)$$

yields an approximate **linearized discrete-time LTI state-space model** about \vec{x} and \vec{u} as

$$\begin{aligned} \Delta \vec{x}[k+1] &\approx F \Delta \vec{x}[k] + G \Delta \vec{u}[k] \\ \Delta \vec{y}[k] &\approx H \Delta \vec{x}[k] + D \Delta \vec{u}[k] \end{aligned} \quad (5.15)$$

Difference Equations and z -Transform

Analogous to continuous-time differential equations for LTI SISO systems, discrete-time LTI SISO systems can use **difference equations** whose standard LTI form is

$$\begin{aligned} y[k] + a_{m-1}y[k-1] + \cdots + a_1y[k-m+1] + a_0y[k-m] \\ = b_p u[k] + b_{p-1}u[k-1] + \cdots + b_1u[k-p+1] + b_0u[k-p] \end{aligned} \quad (5.16)$$

Similarly, the Laplace transform for representing ODEs as continuous-time transfer functions is analogous to the **z -transform** for representing difference equations as a transfer function. For a discrete-time signal $x[k]$, the z -transform is given by

$$x(z) = \mathcal{Z}\{x[k]\} = \sum_{k=-\infty}^{\infty} x[k]z^{-k} \quad (5.17)$$

where z is a complex number. This also has an inverse z -transform

$$x[k] = \mathcal{Z}^{-1}\{X(z)\} = \frac{1}{2\pi j} \oint_C x(z)z^{k-1} dz \quad (5.18)$$

where C is a counterclockwise closed path encircling the origin and entirely in the **region of convergence (ROC)**, i.e. the set of points in the complex plane for which the z -transform summation converges. Thus, for a m^{th} order LTI difference equation, the **discrete-time transfer functions** is

$$H(z) = \frac{y(z)}{u(z)} = \frac{b_p z^p + b_{p-1} z^{p-1} + \cdots + b_1 z + b_0}{z^m + a_{m-1} z^{m-1} + \cdots + a_1 z + a_0} \quad (5.19)$$

For reference, some functions in the k and z domains are provided in the following table.

| Function | k Domain ($\forall k \geq 0$) | z Domain |
|--------------|-----------------------------------|---|
| Unit Step | 1 | $\frac{z}{z-1}$ |
| Power | a^k | $\frac{z}{z-a}$ |
| Sine | $\sin(\omega k)$ | $\frac{z \sin \omega}{z^2 - 2z \cos \omega + 1}$ |
| Cosine | $\cos(\omega k)$ | $\frac{z^2 - z \cos \omega}{z^2 - 2z \cos \omega + 1}$ |
| Power Sine | $a^k \sin(\omega k)$ | $\frac{z^2 - z \cos \omega}{z^2 - 2az \cos \omega + a^2}$ |
| Power Cosine | $a^k \cos(\omega k)$ | $\frac{az \sin \omega}{z^2 - 2az \cos \omega + a^2}$ |

One way to describe the similarity between the s and z transforms is to see that as the s variable operates in multiplication and division as a differentiator and integrator in continuous-time, respectively, the z variable operates in multiplication and division as a backward and forward **time shift operator** in discrete-time, respectively.

For the frequency response of a discrete-time system, one substitutes $z = e^{j\omega}$. Thus, the **discrete-time Fourier transform (DTFT)** with periodicity of 2π can be written as

$$H(\omega) = \sum_{k=-\infty}^{\infty} h[k]e^{-j\omega k} \quad (5.20)$$

and the inverse discrete-time Fourier transform

$$H[k] = \frac{1}{2\pi j} \int_{-\pi}^{\pi} h(e^{j\omega})e^{j\omega k} d\omega \quad (5.21)$$

where the substitution $z = e^{j\omega}$ has set the closed path C to the unit circle. It should be noted that the **discrete Fourier transform (DFT)** discretely samples the continuous DTFT, i.e.

$$H[n] = \frac{1}{N} \sum_{k=0}^{N-1} h[k]e^{-j\frac{2\pi n}{N}k} \quad (5.22)$$

where N is the number of evenly-spaced samples at some **sampling interval** is Δt . The **total sampling time** is $T_s = N\Delta t$, the **sampling frequency** in Hz is $f_s = \frac{1}{\Delta t} = \frac{N}{T_s}$, and the **frequency resolution** is $\Delta f = \frac{f_s}{N}$. Lastly, the **Nyquist frequency**, f_N , defined as

$$f_N = \frac{f_s}{2} = \frac{1}{2\Delta t} = \frac{N}{2T_s} \quad (5.23)$$

is the highest frequency component that can be resolved. This property of the Nyquist frequency leads us to **Shannons sampling theorem** which states that a continuous signal must be discretely sampled at least twice the frequency of the highest frequency in the signal. In this way, one can use the z-transform and apply SISO loop-shaping methods based on discrete-time frequency response requirements. One of the most important algorithms of the twentieth century is the **fast Fourier transform (FFT)** which efficiently computes the DFT.

General Solution of Discrete-Time LTI Systems

Consider the state equation for the discrete-time linear state-space model

$$\vec{x}[k+1] = F_k \vec{x}[k] + G_k \vec{u}[k] \quad (5.24)$$

with a given initial state, $\vec{x}[0] = 0$. Then, iterating through this equation

$$\begin{aligned} \vec{x}[1] &= F_0 \vec{x}[0] + G_0 \vec{u}[0] = G_0 \vec{u}[0] + F_0 \vec{x}[0] \\ \vec{x}[2] &= F_1 \vec{x}[1] + G_1 \vec{u}[1] = G_1 \vec{u}[1] + F_1 G_0 \vec{u}[0] + F_1 F_0 \vec{x}[0] \\ \vec{x}[3] &= F_2 \vec{x}[2] + G_2 \vec{u}[2] = G_2 \vec{u}[2] + F_2 G_1 \vec{u}[1] + F_2 F_1 G_0 \vec{u}[0] + F_2 F_1 F_0 \vec{x}[0] \\ &\vdots \\ \vec{x}[k] &= G_{k-1} \vec{u}[k-1] + F_{k-1} G_{k-2} \vec{u}[k-2] + \dots + F_{k-1} \cdots F_1 G_0 \vec{u}[0] + F_{k-1} \cdots F_0 \vec{x}[0] \end{aligned} \quad (5.25)$$

This general solution can be rewritten as product of two matrices:

$$\vec{x}[k] = F_{k-1} \cdots F_0 \vec{x}[0] + [G_{k-1} \quad F_{k-1}G_{k-2} \quad \cdots \quad F_{k-1} \cdots F_1 G_0] \begin{bmatrix} \vec{u}[k-1] \\ \vec{u}[k-2] \\ \vdots \\ \vec{u}[0] \end{bmatrix} \quad (5.26)$$

which has the form

$$\vec{x}[k] = \Phi(0, k) \vec{x}[0] + C \vec{u} \quad (5.27)$$

where $\Phi(0, k)$ is the **discrete-time state-transition matrix** and C is the discrete-time **state controllability matrix** and has the same structure as continuous-time. Note that for a discrete-time LTI system, one has

$$\Phi(0, k) = F^k \quad (5.28)$$

and

$$C = [G \quad FG \quad \cdots \quad F^k G] \quad (5.29)$$

Similar to the Laplace transform, one can use the Z-transform to obtain the discrete frequency solution from the LTI state equation as

$$z\vec{x}(z) - z\vec{x}[0] = F\vec{x}(z) + G\vec{u}(z) \quad (5.30)$$

or

$$(zI - F)\vec{x}(z) = z\vec{x}[0] + G\vec{u}(z) \quad (5.31)$$

which yields

$$\vec{x}(z) = z(zI - F)^{-1}\vec{x}[0] + (zI - F)^{-1}G\vec{u}(z) \quad (5.32)$$

With the output equation

$$\vec{y}(z) = H\vec{x}(z) + D\vec{u}(z) \quad (5.33)$$

or

$$\vec{y}(z) = Hz(zI - F)^{-1}\vec{x}[0] + \left(H(zI - F)^{-1}G + D\right)\vec{u}(z) \quad (5.34)$$

Discrete-Time State-Space System Stability

For an initial condition at $\vec{x}[0]$ and free response dynamics (i.e. no control), the discrete-time state-space can be reduced to

$$\vec{x}[k] = F\vec{x}[k-1] \quad (5.35)$$

which has the simple solution for the state at time step k as

$$\vec{x}[k] = F^k \vec{x}[0] \quad (5.36)$$

Next, define the **modal state vector**, \vec{z} , such that

$$\vec{x} = V\vec{z} = \vec{v}_1 z_1(t) + \cdots + \vec{v}_{n_x} z_{n_x}(t) \quad (5.37)$$

where V is the matrix of n eigenvectors of F and

$$\vec{z} = V^{-1} \vec{x} = \begin{bmatrix} \vec{\mu}_1^T \vec{x} \\ \vdots \\ \vec{\mu}_{n_x}^T \vec{x} \end{bmatrix} \quad (5.38)$$

where notably, $z_{n_x}(t)$, is dimensionless. Then, substituting into the state equation, one has

$$V \vec{z}[k+1] = FV \vec{z}[k] + G \vec{u}(t) \quad (5.39)$$

or

$$\vec{z}[k+1] = V^{-1} F V \vec{z}[k] + V^{-1} G \vec{u}[k] \quad (5.40)$$

Thus, one can form the discrete-time **Jordan canonical form (JCF)** of the state-space system as

$$\begin{aligned} \vec{z}[k+1] &= \Lambda \vec{z}[k] + V^{-1} G \vec{u}[k] \\ \vec{y}[k] &= G V \vec{z}[k] + D \vec{u}[k] \end{aligned} \quad (5.41)$$

where the state-transition matrix for diagonalizable F is now

$$\Lambda^k = \begin{bmatrix} \lambda_1^k & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \lambda_{n_x}^k \end{bmatrix} \quad (5.42)$$

and this model explicitly **decouples** the state equation into n_x LTI ODEs, i.e.

$$z_i[k] = \lambda_i^k z_i(t) + \vec{\mu}_i^T G \vec{u}(t), \quad i = 1, \dots, n_x \quad (5.43)$$

Consider the zero-input response where the n_x modal states are given by

$$z_i[k] = \lambda_i^k z_i[0] = \lambda_i^k \vec{\mu}_i^T \vec{x}[0] \quad (5.44)$$

and the output response is given by

$$\vec{y}[k] = H(\vec{v}_1 z_1[k] + \cdots + \vec{v}_{n_x} z_{n_x}[k]) \quad (5.45)$$

Thus, one can see the modal states are characterized by the eigenvalues and left eigenvectors while the output response is additionally characterized by the right eigenvectors and the modal states. As these eigenvalues will occur as real numbers or in complex-conjugate pairs, the modal states may be complex-valued which describes the oscillatory nature of the modal responses. Regardless, the magnitude of each individual mode allows one to characterize the stability of each mode as well as the discrete-time LTI system as a whole. In particular, a mode is **stable** if $|\lambda_i| < 0$, **marginally stable** if $|\lambda_i| = 0$, and **unstable** if $|\lambda_i| > 0$. Furthermore, the LTI system is **stable** if *all* the modes are strictly stable, the LTI system is **marginally stable** if *all* the modes are stable or marginally stable, and if *any* mode is unstable, the LTI system is **unstable**. As these eigenvalues may be complex-valued, one can also state that all the system poles located in the left half of the complex plane, i.e. the **left half plane (LHP)**, correspond to stable modes.

Discretization Techniques

Often one wishes to discretize a continuous-time LTI system, i.e. convert it to a discrete-time LTI system. To discretize general MIMO systems, recall the continuous-time state equation

$$\dot{\vec{x}}(t) = A\vec{x}(t) + B\vec{u}(t) \quad (5.46)$$

which has the general solution

$$\vec{x}(t) = e^{At}\vec{x}(0) + \int_0^t e^{A(t-\tau)}B\vec{u}(\tau)d\tau \quad (5.47)$$

Thus, for every time step from $k - 1$ to k and $t = \Delta t$, one can write

$$\vec{x}[k] = e^{A\Delta t}\vec{x}[k - 1] + \int_0^{\Delta t} e^{A(\Delta t-\tau)}B\vec{u}(\tau)d\tau \quad (5.48)$$

which infers that

$$F = e^{A\Delta t} \quad (5.49)$$

Note that this explains the connection between the LTI stability conditions on F and the corresponding LTI stability condition on A , i.e. magnitude of eigenvalues of F are less than one and the real part of the eigenvalues of A are negative.

However, for the input matrix, one must assume that $\vec{u}(\tau)$ follows some type of piecewise function between times steps. To only use the previous time step to approximate the integral, one may use a **zero-order hold (ZOH)** which assumes $\vec{u}(\tau)$ is a piecewise constant input $\vec{u}[k - 1]$ from $k - 1$ to k , then

$$\int_0^{\Delta t} e^{A(\Delta t-\tau)}B\vec{u}(\tau)d\tau = \left(\int_0^{\Delta t} e^{A(\Delta t-\tau)}Bd\tau \right) \vec{u}[k - 1] \quad (5.50)$$

and one has for the discrete-time input matrix

$$G = \int_0^{\Delta t} e^{A(\Delta t-\tau)}B(\tau)d\tau \quad (5.51)$$

Other alternatives include higher-order holds, including a **first-order hold (FOH)** which assumes $\vec{u}(\tau)$ is a piecewise linear input from $k - 1$ to k . However, other holds require knowledge of the inputs at additional time steps, e.g. FOH requires $u[k]$ and $u[k - 1]$. Lastly, note that $H = C$ in this discretization.

Analogous to the ZOH and FOH, one can also approximate the transfer functions of SISO systems, using the **Euler Forward approximation**, i.e.

$$s \approx \frac{z - 1}{\Delta t} \quad (5.52)$$

or the **Tustin approximation**, i.e.

$$s \approx \frac{2(z - 1)}{\Delta t(z + 1)} \quad (5.53)$$

and another common method is the **Euler Backward approximation**, i.e.

$$s \approx \frac{z - 1}{z\Delta t} \quad (5.54)$$

which is numerically more stable than the forward version due to the backward requiring smaller sampling time Δt . Even so, using these latter two approximation methods to convert controllers typically requires that the sampling time satisfy

$$\Delta t < \frac{\pi}{5\omega_c} \quad (5.55)$$

in order for the controller to *likely* produce satisfactory closed-loop behavior where ω_c is the crossover frequency of the open-loop transfer function.

References

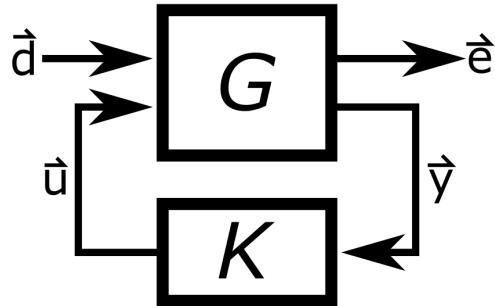
For more information, please refer to the following

- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “7 Digital Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 584-620

5.2 Discrete-Time Linear Feedback Control Systems

Discrete-Time Generalized LTI Feedback Control System

Consider the generalized LTI feedback control system architecture



where the vector input, $\vec{d} \in \mathbb{R}^{n_d}$, to the generalized plant is known as the **generalized disturbance** which typically includes reference commands, \vec{r} , process noise, \vec{w} , and measurement noise, \vec{v} . The vector output, $\vec{e} \in \mathbb{R}^{n_e}$, from the generalized plant is known as the **generalized error** which typically includes at least the weighted tracking error and the weighted control effort. The other vectors are the **generalized control input** vector $\vec{u} \in \mathbb{R}^{n_u}$ and the **generalized output!discrete-time**, $\vec{y} \in \mathbb{R}^{n_y}$.

Just as for continuous-time, the LTI state-space model for the generalized plant, G , assuming **loop-shifting**, can be written as

$$\begin{aligned} \vec{x}[k+1] &= F\vec{x}[k] + [G_1 \quad G_2] \begin{bmatrix} \vec{d}[k] \\ \vec{u}[k] \end{bmatrix} \\ \begin{bmatrix} \vec{e}[k] \\ \vec{y}[k] \end{bmatrix} &= \begin{bmatrix} H_1 \\ H_2 \end{bmatrix} \vec{x}[k] + \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & 0 \end{bmatrix} \begin{bmatrix} \vec{d}[k] \\ \vec{u}[k] \end{bmatrix} \end{aligned} \quad (5.56)$$

with the **generalized LTI feedback controller**, K , designed as

$$\begin{aligned}\vec{x}_K[k+1] &= F_K \vec{x}_K[k] + G_K \vec{y}[k] \\ \vec{u}[k] &= H_K \vec{x}_K[k] + D_K \vec{y}[k]\end{aligned}\quad (5.57)$$

and the interconnection of G and K is well-posed if and only if $(I - D_{22}D_K)^{-1}$ exists. In the same way, one can obtain the discrete-time closed-loop LTI state-space system model as

$$\begin{aligned}\begin{bmatrix} \vec{x}[k+1] \\ \vec{x}_K[k+1] \end{bmatrix} &= \begin{bmatrix} F + G_2 D_K H_2 & G_2 H_K \\ G_K H_2 & F_K \end{bmatrix} \begin{bmatrix} \vec{x}[k] \\ \vec{x}_K[k] \end{bmatrix} + \begin{bmatrix} G_1 + G_2 D_K D_{21} \\ G_K D_{21} \end{bmatrix} \vec{d}[k] \\ \vec{e}[k] &= [H_1 + D_{12} D_K H_2 \quad D_{12} H_K] \begin{bmatrix} \vec{x}[k] \\ \vec{x}_K[k] \end{bmatrix} + (D_{11} + D_{12} D_K D_{21}) \vec{d}[k]\end{aligned}\quad (5.58)$$

with the definitions for the closed-loop state matrix, F_L , as

$$F_L = \begin{bmatrix} F + G_2 D_K H_2 & G_2 H_K \\ G_K H_2 & F_K \end{bmatrix} \quad (5.59)$$

the closed-loop input matrix, G_L , as

$$G_L = \begin{bmatrix} G_1 + G_2 D_K D_{21} \\ G_K D_{21} \end{bmatrix} \quad (5.60)$$

the closed-loop output matrix, H_L , as

$$H_L = [H_1 + D_{12} D_K H_2 \quad D_{12} H_K] \quad (5.61)$$

and the closed-loop feedthrough matrix, D_L , as

$$D_L = D_{11} + D_{12} D_K D_{21} \quad (5.62)$$

which for stability of the generalized feedback control system, one requires that F_L is stable.

Discrete-Time Output and State Feedback Control

If $F_K = G_K = H_K = 0$, one has **static-controller feedback**, a type of **fixed-gain controller**. The general case is also known as a.k.a. **output feedback control**, where one has an output feedback controller as

$$\vec{u}[k] = D_K \vec{y}[k] \quad (5.63)$$

which results in a simplified closed-loop LTI state-space model

$$\begin{aligned}\vec{x}[k+1] &= F_L \vec{x}[k] + G_L \vec{d}[k] \\ \vec{e}[k] &= H_L \vec{x}[k] + D_L \vec{d}[k]\end{aligned}\quad (5.64)$$

which results in a closed-loop state matrix, F_L , as

$$F_L = F + G_2 D_K H_{12} \quad (5.65)$$

a closed-loop input matrix, G_L , as

$$G_L = G_1 + G_2 D_K D_{21} \quad (5.66)$$

a closed-loop output matrix, H_L , as

$$H_L = H_1 + D_{12} D_K H_2 \quad (5.67)$$

and a closed-loop feedthrough matrix, D_L , as

$$D_L = D_{11} + D_{12} D_K D_{21} \quad (5.68)$$

Thus, an output feedback control system is stable if and only if $F + G_2 D_K H_{12}$ is stable.

A special case of fixed-gain feedback control is **state feedback control**, i.e. $H_2 = I$, $D_{21} = 0$, and state feedback controller

$$\vec{u}[k] = D_K \vec{x}[k] \quad (5.69)$$

with closed-loop state matrix, F_L , is

$$F_L = F + G_2 D_K \quad (5.70)$$

the closed-loop input matrix, G_L , is

$$G_L = G_1 \quad (5.71)$$

the closed-loop output matrix, H_L , as

$$H_L = H_1 + D_{12} D_K \quad (5.72)$$

and the closed-loop feedthrough matrix, D_L , as

$$D_L = D_{11} \quad (5.73)$$

Thus, a state feedback control system is stable if and only if $F + G_2 D_K$ is stable.

Discrete-Time Observer Feedback Control

The general case of **dynamic-controller feedback** is also known as discrete-time **observer feedback control**, one defines the controller state as the **state estimate**, $\hat{\vec{x}}$, i.e.

$$\vec{x}_K[k] = \hat{\vec{x}}[k] \quad (5.74)$$

where an observer is designed to form $\hat{\vec{x}}[k]$ and uses this to form the control input as

$$\vec{u}[k] = -K \hat{\vec{x}}[k] \quad (5.75)$$

Here, an **open-loop observer** could be formed based on the linear state-space model for continuous-time, assuming the disturbances \vec{d} are unknown, as

$$\hat{\vec{x}}[k+1] = F \hat{\vec{x}}[k] + G_2 \vec{u}[k] \quad (5.76)$$

However, for feedback control, one receives the output signal from the output equation, and forms a **closed-loop observer** for discrete-time as

$$\begin{aligned} \hat{\vec{x}}[k+1] &= F \hat{\vec{x}}(t) + G_2 \vec{u}[k] + L \left(\vec{y}[k] - \hat{\vec{y}}[k] \right) \\ \hat{\vec{y}}[k] &= H_2 \hat{\vec{x}}[k] \end{aligned} \quad (5.77)$$

where $\hat{\vec{y}}$ is the output estimate based on the output equation model and L is the **Luenberger observer matrix**. Thus, with $\vec{u}[k] = -K\hat{\vec{x}}[k]$, one can form the continuous-time **observer feedback control system** as the generalized LTI feedback controller

$$\begin{aligned}\hat{\vec{x}}[k+1] &= (F - G_2K - LH_2) \hat{\vec{x}}[k] + L\vec{y}(t) \\ \vec{u}[k] &= -K\hat{\vec{x}}(t)\end{aligned}\quad (5.78)$$

where

$$F_K = F - G_2K - LH_2 \quad (5.79)$$

$$G_K = L \quad (5.80)$$

$$H_K = -K \quad (5.81)$$

$$D_K = 0 \quad (5.82)$$

Thus, one has the closed-loop dynamics

$$\begin{aligned}\begin{bmatrix} \vec{x}[k+1] \\ \hat{\vec{x}}[k+1] \end{bmatrix} &= \begin{bmatrix} F & -G_2K \\ LH_2 & F - G_2K - LH_2 \end{bmatrix} \begin{bmatrix} \vec{x}[k] \\ \hat{\vec{x}}[k] \end{bmatrix} + \begin{bmatrix} G_1 \\ LD_{21} \end{bmatrix} \vec{d}[k] \\ \vec{e}[k] &= [H_1 \quad -D_{12}K] \begin{bmatrix} \vec{x}[k] \\ \hat{\vec{x}}[k] \end{bmatrix} + D_{11}\vec{d}[k]\end{aligned}\quad (5.83)$$

However, to better assess the stability of the closed-loop system, consider the **state error**, $\vec{e}_x[k]$, defined as

$$\vec{e}_x[k] = \vec{x}[k] - \hat{\vec{x}}[k] \quad (5.84)$$

Then, after the same algebra as for continuous-time, one has

$$\begin{aligned}\begin{bmatrix} \vec{x}[k+1] \\ \vec{e}_x[k+1] \end{bmatrix} &= \begin{bmatrix} (F - G_2K) & G_2K \\ 0 & F - LH_2 \end{bmatrix} \begin{bmatrix} \vec{x}[k] \\ \vec{e}_x[k] \end{bmatrix} + \begin{bmatrix} G_1 \\ G_1 - LD_{21} \end{bmatrix} \vec{d}[k] \\ \vec{e}[k] &= [(H_1 - D_{12}K) \quad D_{12}K] \begin{bmatrix} \vec{x}[k] \\ \vec{e}_x[k] \end{bmatrix} + D_{11}\vec{d}[k]\end{aligned}\quad (5.85)$$

Thus, a discrete-time observer feedback control system is stable if and only if $F - G_2K$ and $F - LH_2$ are stable, i.e., one has the same **separation principle** for discrete-time observer feedback control.

Lastly, it should be noted that the Luenberger observer is also known as an $\alpha - \beta$ **filter** if

1. F
 $mathbb{R}^{2 \times 2}$ with all main diagonal terms equal to 1 and upper diagonal terms equal to Δt .
2. $H = [10]$
3. $L = [\alpha\beta]^T$
4. u is known or set to zero.
5. G has non-zero gain term as last element if u is non-zero.

Controllability and Observability of Discrete-Time Linear Systems

Recall the discrete-time **state controllability matrix**

$$C = \begin{bmatrix} G_{k-1} & F_{k-1}G_{k-2} & \cdots & F_{k-1}\cdots F_1G_0 \end{bmatrix} \quad (5.86)$$

and for LTI system, one has

$$C = \begin{bmatrix} G & FG & \cdots & F^k G \end{bmatrix} \quad (5.87)$$

which has the same structure as continuous-time. Thus, the same logic applies that any $\vec{x}[k]$ can be reached if and only if C has full rank. Similarly one can define the **controllability Gramian** for discrete-time as

$$W_C[k] = \sum_{k=0}^{\infty} \Phi(0, k) G_k G_k^T \Phi(0, k)^T \quad (5.88)$$

where one can show if and only if $W[k]$ full rank for *any* $t > 0$ or $k > 0$, then the system is state controllable. Note that W_C is $n_x \times n_x$ matrix, thus the degree of controllability possible by assessing eigenvalue decomposition of W_C .

State observability is defined as: if, for some finite $N > 0$, inputs $\vec{u}[k]$, and outputs $\vec{y}[k]$ with $0 < k \leq N$, the initial state $\vec{x}[0]$ can be determined, then one can observe the system's *past* initial state. One can quantify this concept without loss of generality given the input and output sequences, $\vec{u}[k]$ and $\vec{y}[k]$ for $k = 0, 1, \dots, n - 1$, and the discrete-time linear output equation

$$\vec{y}[k] = H_k \vec{x}[k] + D_k \vec{u}[k] \quad (5.89)$$

which can be rearranged as

$$\vec{y}[k] - D_k \vec{u}[k] = H_k \vec{x}[k] \quad (5.90)$$

Thus, one has the following sequence of output equations

$$\vec{y}[0] - D_0 \vec{u}[0] = H_0 \vec{x}[0] \quad (5.91)$$

$$\vec{y}[1] - D_1 \vec{u}[1] = H_1 \vec{x}[1] = H_1 (F_0 \vec{x}[0] + G_0 \vec{u}[0]) = H_1 F_0 \vec{x}[0] + H_1 G_0 \vec{u}[0] \quad (5.92)$$

and

$$\vec{y}[2] - D_2 \vec{u}[2] = H_2 \vec{x}[2] = H_2 (F_1 \vec{x}[1] + G_1 \vec{u}[1]) \quad (5.93)$$

or more simply

$$\vec{y}[2] - D_2 \vec{u}[2] = H_2 F_1 (F_0 \vec{x}[0] + G_0 \vec{u}[0]) + H_2 G_1 \vec{u}[1] \quad (5.94)$$

$$\vec{y}[2] - D_2 \vec{u}[2] = H_2 F_1 F_0 \vec{x}[0] + H_2 F_1 G_0 \vec{u}[0] + H_2 G_1 \vec{u}[1] \quad (5.95)$$

Thus, by extension one has

$$\vec{y}[n-1] - D_{n-1} \vec{u}[n-1] = H_{n-1} F_{n-2} \cdots F_0 \vec{x}[0] + H_{n-1} F_{n-2} \cdots F_1 G_0 \vec{u}[0] + \cdots H_{n-1} G_{n-2} \vec{u}[n-2] \quad (5.96)$$

and rearranging as

$$\begin{bmatrix} \vec{y}[0] \\ \vec{y}[1] \\ \vdots \\ \vec{y}[n-2] \\ \vec{y}[n-1] \end{bmatrix} - \begin{bmatrix} D_0 & 0 & \cdots & 0 & 0 \\ HG_0 & D_1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ HF^{n-3}G & HF^{n-4}G & \cdots & D_{n-2} & 0 \\ H_{n-1}F^{n-2}G_0 & H_{n-1}F^{n-3}G_1 & \cdots & H_{n-1}G_{n-2} & D_{n-1} \end{bmatrix} \begin{bmatrix} \vec{u}[0] \\ \vec{u}[1] \\ \vdots \\ \vec{u}[n-2] \\ \vec{u}[n-1] \end{bmatrix} = \begin{bmatrix} H_0 \\ H_1F_0 \\ \vdots \\ H_{n-1}F_{n-2} \cdots F_0 \end{bmatrix} \vec{x}[0] \quad (5.97)$$

which has the form

$$\vec{y} = \mathcal{H}\vec{x}[0] \quad (5.98)$$

where \mathcal{H} is the **observability matrix**. Thus, by inspection one can see that *any* $\vec{x}[0]$ can be observed if and only if **observability matrix** \mathcal{H} has full row rank, i.e. $\text{rank } \mathcal{H} = n_x$. The initial state can be computed by

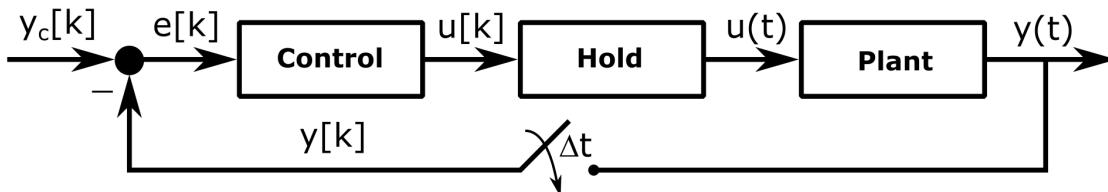
$$\vec{x}(0) = (\mathcal{H}^T \mathcal{H})^{-1} \mathcal{H}^T \vec{y} \quad (5.99)$$

where $(\mathcal{H}^T \mathcal{H})^{-1} \mathcal{H}^T$ is known as the **pseudoinverse** of \mathcal{H} which exists if \mathcal{H} has full row rank.

Hybrid-Time Feedback Control Systems

Before modern computers, flight controller design was performed using standard circuit components to implement continuous-time control laws, i.e. “classical” control. However, modern flight control systems are implemented by microprocessors or digital computers where one has a digital control system where time and values are discrete. Thus, there are additional considerations that must be taken for hardware implementation of the control system concepts discussed in this textbook. The additional effects of discrete-values in truly digital systems is beyond the scope of this textbook, but generally has little effect if the numerical precision of the digital computer is high enough relative to the system dynamics.

One approach is to use the continuous-time control design methods in this course and then convert the continuous-time control system to a discrete-time control system, i.e., **discretization**. However, one may also directly design the discrete-time feedback control system by modeling the discrete-time effects with the continuous-time plant while using discrete-time control methods for the controller. This is called a **hybrid-time feedback control system**. In general, the block diagram for a **hybrid-time feedback control system** can be considered as the following.



This system notably consists of the following hybrid processing steps in addition to the continuous-time plant.

- **Sampling:** The output of the system, $y(t)$ is updated in the digital system every Δt forming the k^{th} sampled output, $y[k]$, where $k = 1, 2, \dots$ is the time step, i.e. $t = k\Delta t$. This is typically performed by an analog-to-digital converter (ADC).
- **Control Update:** The control input at time step k , $u[k]$, is updated using a digital computer. This is typically performed by a microprocessor with a “real-time” operating system (RTOS).
- **Hold:** The updated control input, $u[k]$, is converted into a continuous-time signal $u(t)$ for the physical (e.g. electro-mechanical) system. This is typically performed by a digital-to-analog converter (DAC).

It should be noted that the signal processing from system to system will also cause some small time delays that should be modeled in the full system analysis and typically would also include an actuator in the plant model. Furthermore, if one is simply discretizing a continuous-time control law to discrete-time without this additional analysis, one must ensure that the Nyquist frequency, $\Delta t/2$, is much faster than any relevant dynamics. However, if this is not that case, one must consider the real continuous-time effects of the digital feedback control system primarily including the sampling and hold steps.

5.3 Discrete-Time Linear-Quadratic Regulator

Discrete-Time Systems and Dynamic Programming

The OCP for discrete-time dynamical systems can be generally written as

$$\vec{u}^{\text{opt}}[k] = \underset{\vec{u}[k] \text{ for } k=0, \dots, N-1}{\operatorname{argmin}} \quad \mathcal{J} = \mathcal{E}(\vec{x}[N], N) + \sum_{k=0}^{N-1} \mathcal{L}(\vec{x}[k], \vec{u}[k], k)$$

subject to:

discrete dynamics: $\vec{x}[k+1] = f(\vec{x}[k], \vec{u}[k], k)$
initial condition: $\vec{x}[0] = \vec{x}_0$
constraints: $c(\vec{x}[k], \vec{u}[k], k) \leq 0$

(5.100)

where $\vec{u}^{\text{opt}}[k]$ is the **optimal control sequence**, N is the final time step, also known as the discrete-time **time horizon** since k starts at zero without loss of generality), and \mathcal{J} is the **objective function**, also known as the **cost function**. Comparing this discrete-time OCP to the continuous-time OCP, the integration became a summation and the differential equation became a difference equation.

A key concept in optimization over time is one can perform the optimization in stages. In essence, one is balancing the lowest possible cost at the present stage against the impact this would have for costs at future stages. The optimal control action minimizes the sum of the cost incurred at the current stage and the least total cost that can be incurred from all subsequent stages, consequent on this decision. This is known as the **principle of optimality** which states that from any point on an optimal trajectory, the remaining trajectory is optimal for the corresponding problem initiated at that point. Formally for discrete-time OCPs, this can be stated by defining the **cost-to-go function** at time step k as

$$\mathcal{J}_k(\vec{x}[k]) = \mathcal{E}(\vec{x}[N]) + \sum_{i=k}^{N-1} \mathcal{L}(\vec{x}[i], \vec{u}[i]) \quad (5.101)$$

which has a corresponding sub-optimization problem for the minimum cost-to-go, $\mathcal{V}_k(\vec{x}[k])$, also known as the **value function**, as

$$\mathcal{V}_k(\vec{x}[k]) = \min_{\vec{u}[i] \text{ for } i=k, \dots, N-1} \mathcal{J}_k = \mathcal{E}(\vec{x}[N], N) + \sum_{i=k}^{N-1} \mathcal{L}(\vec{x}[i], \vec{u}[i]) \quad (5.102)$$

subject to the dynamics, initial conditions, and constraints.

Considering \mathcal{V}_k as a function of k and $\vec{x}[k]$ with the following properties

$$\mathcal{V}_N(\vec{x}[N]) = \mathcal{E}(\vec{x}[N], N) \quad (5.103)$$

and

$$\mathcal{V}_0(\vec{x}[0]) = \min_{\vec{u}[k] \text{ for } k=0, \dots, N-1} \mathcal{J} \quad (5.104)$$

Furthermore, this formalization allows one to write the sub-optimization as a recursive relationship

$$\mathcal{V}_k(\vec{x}[k]) = \min_{\vec{u}[i] \text{ for } i=k, \dots, N-1} \mathcal{L}(\vec{x}[k], \vec{u}[k]) + \mathcal{V}_{k+1}(\vec{x}[k+1]) \quad (5.105)$$

which is the OCP form of the **Bellman equation**, also known as the **optimality equation**. Thus, one can begin at $\mathcal{V}_N(\vec{x}[N])$ and calculate the value function at previous time steps by working backwards using the Bellman equation and finally obtaining $\mathcal{V}_0(\vec{x}[0])$ as the value of the optimal cost. Then, the optimal control sequence, $\vec{u}^{\text{opt}}[k]$, can then be recovered by tracing back the calculations already performed for the value function.

The Bellman equation serves as the fundamental result of dynamic programming (DP) and is often referred to as the **dynamic programming equation**. By definition, **dynamic programming (DP)** solves an optimization problem by recursively solving simpler sub-problems which make up the overall problem. In general, not all optimization problems allow dynamic programming methods, however optimization problems over a sequence of time steps often allow recursive sub-problems to be nested inside the overall problem. This time sequencing is the origin of the term “dynamic” in dynamic programming. Mathematically, these sub-problems done by defining a sequence of value functions, as shown previously for discrete-time OCPs. The value function at any time step is valued based on future time steps and thus can be used to compute the minimum cost-to-go function.

Unconstrained Discrete-Time LQR

The **unconstrained finite-horizon discrete-time LQ OCP** can be stated as

$$\begin{aligned} \vec{u}^{\text{opt}}[k] = \underset{\vec{u}[k] \text{ for } k=0, \dots, N-1}{\text{argmin}} \quad & \mathcal{J} = \vec{x}[N]^T E \vec{x}[N] + \sum_{k=0}^{N-1} \vec{x}[k]^T Q \vec{x}[k] + \vec{u}[k]^T R \vec{u}[k] + 2 \vec{x}[k]^T S \vec{u}[k] \\ \text{subject to: } & \vec{x}[k+1] = F \vec{x}[k] + G \vec{u}[k] \\ \text{initial condition: } & \vec{x}[k] = \vec{x}_0 \end{aligned} \quad (5.106)$$

where the unconstrained finite-horizon discrete-time LQR is the optimal control sequence, $\vec{u}^{\text{opt}}[k]$, which minimizes the quadratic cost function, \mathcal{J} .

Recalling the dynamic programming solution method, one can define the following value function for this discrete-time OCP:

$$\mathcal{V}_k(\vec{x}[k]) = \min_{\vec{u}[k] \text{ for } k=0, \dots, N-1} \vec{x}_N^T E \vec{x}_N + \sum_{\tau=k}^{N-1} \vec{x}_{\tau}^T Q \vec{x}_{\tau} + \vec{u}_{\tau}^T R \vec{u}_{\tau} + 2 \vec{x}_{\tau}^T S \vec{u}_{\tau} \quad (5.107)$$

which has the boundary condition

$$\mathcal{V}_k(\vec{x}) = \vec{x}^T E \vec{x} \quad (5.108)$$

Moreover, the incurred cost for the LQ OCP can be defined as

$$\mathcal{L}(\vec{x}[k], \vec{u}[k]) = \vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u} \quad (5.109)$$

and the cost-to-go from time step k due to “next” state governed by the linear dynamics of the LQR OCP can be identified as

$$\mathcal{V}_{k+1}(\vec{x}[k+1]) = \mathcal{V}_{k+1}(F \vec{x}[k] + G \vec{u}[k]) \quad (5.110)$$

which, by the Bellman equation, one has

$$\mathcal{V}_k(\vec{x}[k]) = \min_{\vec{u}} \left(\vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u} + 2 \vec{x}^T S \vec{u} + \mathcal{V}_{k+1}(F \vec{x} + G \vec{u}) \right) \quad (5.111)$$

To solve this OCP, assume the value function has a quadratic form, i.e.

$$\mathcal{V}_k(\vec{x}) = \vec{x}^T P[k] \vec{x} \quad (5.112)$$

with $P[k]$ being a symmetric and positive semi-definite matrix and with the condition

$$P[N] = E \quad (5.113)$$

and by substitution into the Bellman equation, one has

$$\mathcal{V}_{k-1}(\vec{x}) = \vec{x}^T Q \vec{x} + \vec{u}^T R \vec{u} + 2 \vec{x}^T S \vec{u} + (F \vec{x} + G \vec{u})^T P[k] (F \vec{x} + G \vec{u}) \quad (5.114)$$

and setting

$$\frac{d\mathcal{V}_{k-1}}{d\vec{u}} = 0 \quad (5.115)$$

one has

$$2 \vec{u}^{\text{opt}} R + 2 \vec{x}^T S + 2(F \vec{x} + G \vec{u}^{\text{opt}})^T P[k] G = 0 \quad (5.116)$$

Dividing by 2 and rearranging, one has

$$\vec{u}^{\text{opt}} R + \vec{u}^{\text{opt}} T G^T P[k] G = -\vec{x}^T (F^T P[k] G + S) \quad (5.117)$$

and taking the transpose

$$R \vec{u}^{\text{opt}} + G^T P[k] G \vec{u}^{\text{opt}} = -(G^T P[k] F + S^T) \vec{x} \quad (5.118)$$

and the inverse

$$\vec{u}^{\text{opt}} = - \left(G^T P[k] G + R \right)^{-1} \left(G^T P[k] F + S^T \right) \vec{x} \quad (5.119)$$

one obtains the optimal control input for a single time step and can be summarized for entire sequence as

$$\vec{u}^{\text{opt}}[k] = -K[k]x[k] \quad (5.120)$$

with

$$K[k] = \left(G^T P[k] G + R \right)^{-1} \left(G^T P[k] F + S^T \right) \quad (5.121)$$

where one still must solve for $P[k]$ to get an explicit solution.

First, using \vec{u}^{opt} and

$$2\vec{x}^T S \vec{u}^{\text{opt}} = \vec{x}^T S \vec{u}^{\text{opt}} + \vec{u}^{\text{opt}}^T S \vec{x} \quad (5.122)$$

one can rewrite the Bellman equation as

$$\mathcal{V}_{k-1}(\vec{x}) = \vec{x}^T Q \vec{x} + \vec{u}^{\text{opt}}^T R \vec{u}^{\text{opt}} + \vec{x}^T S \vec{u}^{\text{opt}} + \vec{u}^{\text{opt}}^T S \vec{x} + (F \vec{x} + G \vec{u}^{\text{opt}})^T P[k] (F \vec{x} + G \vec{u}^{\text{opt}}) \quad (5.123)$$

and distributing out the quadratic components, one has four terms

$$\begin{aligned} \mathcal{V}_{k-1}(\vec{x}) &= \vec{x}^T \left(Q + F^T P[k] F \right) \vec{x} \\ &\quad + \vec{u}^{\text{opt}}^T \left(G^T P[k] G + R \right) \vec{u}^{\text{opt}} \\ &\quad + \vec{x}^T \left(F^T P[k] G + S \right) \vec{u}^{\text{opt}} \\ &\quad + \vec{u}^{\text{opt}}^T \left(G^T P[k] F + S^T \right) \vec{x} \end{aligned} \quad (5.124)$$

Next, one can substitute for \vec{u}^{opt} from the solution earlier and by using the equivalent

$$\vec{u}^{\text{opt}}^T = -\vec{x}^T \left(F^T P[k] G + S \right) \left(G^T P[k] G + R \right)^{-1} \quad (5.125)$$

the Bellman equation becomes

$$\begin{aligned} \mathcal{V}_{k-1}(\vec{x}) &= \vec{x}^T \left(Q + F^T P[k] F \right) \vec{x} \\ &\quad + \vec{x}^T \left(F^T P[k] G + S \right) \left(G^T P[k] G + R \right)^{-1} \\ &\quad \left(G^T P[k] F + S^T \right) \vec{x} \\ &\quad - \vec{x}^T \left(F^T P[k] G + S \right) \left(G^T P[k] G + R \right)^{-1} \\ &\quad \left(G^T P[k] F + S^T \right) \vec{x} \\ &\quad - \vec{x}^T \left(F^T P[k] G + S \right) \left(G^T P[k] G + R \right)^{-1} \\ &\quad \left(G^T P[k] F + S^T \right) \vec{x} \end{aligned} \quad (5.126)$$

and by simplifying

$$\begin{aligned} \mathcal{V}_{k-1}(\vec{x}) &= \vec{x}^T P[k-1] \vec{x} = \vec{x}^T \left[F^T P[k] F - \left(F^T P[k] G + S \right) \right. \\ &\quad \left. \left(G^T P[k] G + R \right)^{-1} \left(G^T P[k] F + S^T \right) \right] \vec{x} \end{aligned} \quad (5.127)$$

Finally, the quadratic form for the value function provides a recursive equation for $P[k - 1]$ given $P[k]$ as

$$P[k - 1] = F^T P[k] F + Q - \left(F^T P[k] G + S \right) \left(G^T P[k] G + R \right)^{-1} \left(G^T P[k] F + S^T \right) \quad (5.128)$$

which is known as the **dynamic Riccati equation** which is a backwards recursion formula to solve for the discrete-time **Riccati matrix**, $P[k]$, over some time horizon N starting from $P[N] = E$.

Finally, the **unconstrained finite-horizon discrete-time LQR** has the form

$$\vec{u}^{\text{opt}} = -K[k] \vec{x}[k] \quad \forall k \quad (5.129)$$

where

$$K[k] = \left(G^T P[k+1] G + R \right)^{-1} \left(G^T P[k+1] F + S^T \right) \quad (5.130)$$

which notably results in closed-loop state-space dynamics represented by

$$\vec{x}[k+1] = (F - GK[k]) \vec{x}[k] \quad (5.131)$$

Finally, if one considers the **unconstrained infinite-horizon discrete-time LQ OCP**, the solution is given by the steady-state of the dynamic Riccati equation, i.e.

$$P = F^T P F + Q - \left(F^T P G + S \right) \left(G^T P G + R \right)^{-1} \left(G^T P F + S^T \right) \quad (5.132)$$

which is known as the **discrete-time algebraic Riccati equation (DARE)**.

It should be noted that if one has LTV dynamics and/or time-varying cost matrices, the dynamic Riccati equation still applies. However, a steady-state solution will exist if for $N \gg 1$, the difference between $P[N] - P[N-1] \rightarrow 0$ as $t \rightarrow 0$. Otherwise, the discrete-time LTV LQR cannot only be implemented for an infinite-horizon. Typically, in order to accomplish this steady-state, one must design $Q[k]$ and $R[k]$ to have some structure.

5.4 Iterative and Extended Linear-Quadratic Regulator

For nonlinear and/or non-quadratic discrete-time OCPs, one can often form locally-optimal control sequences using approximation methods for the OCP. Iterative LQR (ILQR) and extended LQR (ELQR) are two common LQR-based methods derived that use forwards-backwards iterative procedures with linearization for the dynamics and/or quadratization methods for the cost functions. It should also be pointed out that these methods also appear in the iterative extended Kalman filter and the iterative extended Kalman smoother discussed later in this textbook for nonlinear Bayesian state estimation.

To this end, consider the unconstrained, nonlinear, non-quadratic finite-horizon discrete-time OCP written as

$$\begin{aligned} \vec{u}_k^{\text{opt}} &= \underset{\vec{u}_k \text{ for } k=0, \dots, N-1}{\text{argmin}} \quad \mathcal{J} = \mathcal{E}(\vec{x}_N) + \sum_{k=0}^{N-1} \mathcal{L}_k(\vec{x}_k, \vec{u}_k) \\ &\text{subject to:} \end{aligned} \quad (5.133)$$

discrete dynamics: $\vec{x}_{k+1} = f_k(\vec{x}_k, \vec{u}_k)$

initial condition: \vec{x}_0

where the solution will provide a locally optimal control sequence of the form

$$\vec{u}_k = \pi_k(\vec{x}_k) \quad k = 0, \dots, N-1 \quad (5.134)$$

which has a definable inverse or “backward” control sequence

$$\tilde{\vec{u}}_k = \tilde{\pi}_k(\vec{x}_{k+1}) \quad k = 0, \dots, N-1 \quad (5.135)$$

as opposed to the forward control law.

Consider a nominal trajectory defined as $\hat{\vec{x}}_k$ with $k = 0, \dots, N$ with a nominal control input as $\hat{\vec{u}}_k$ for $k = 0, \dots, N-1$ where the state trajectory is computed via the (forward) dynamics as

$$\hat{\vec{x}}_{k+1} = f_k(\hat{\vec{x}}_k, \hat{\vec{u}}_k) \quad (5.136)$$

and the control input is computed with the current chosen (forward) control law, $\pi_k()$, as

$$\hat{\vec{u}}_k = \pi_k(\hat{\vec{x}}_{k+1}) \quad (5.137)$$

or, alternatively, the state trajectory is computed via the backward dynamics, $\tilde{f}_k()$ as

$$\hat{\vec{x}}_k = \tilde{f}_k(\hat{\vec{x}}_{k+1}, \hat{\vec{u}}_k) \quad (5.138)$$

and the control input is computed with the current chosen backward control law, $\tilde{\pi}_k()$, as

$$\hat{\vec{u}}_k = \tilde{\pi}_k(\hat{\vec{x}}_{k+1}) \quad (5.139)$$

In the I/ELQR approaches, assume one can quadratize the local cost function as

$$\delta \mathcal{J}_k = \frac{1}{2} \vec{x}_N^T E \vec{x}_N + \vec{x}_N^T \vec{e} + \sum_{k=0}^{N-1} \frac{1}{2} \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix} + \begin{bmatrix} \vec{q}_k \\ \vec{r}_k \end{bmatrix}^T \begin{bmatrix} \vec{q}_k \\ \vec{r}_k \end{bmatrix} \quad (5.140)$$

where

$$E = \left. \frac{\partial^2 \mathcal{E}}{\partial \vec{x}_k \partial \vec{x}_k} \right|_{\vec{x}_N = \hat{\vec{x}}_N} \quad (5.141)$$

$$\vec{e} = \left. \frac{\partial \mathcal{E}}{\partial \vec{x}_k} \right|_{\vec{x}_N = \hat{\vec{x}}_N} - E \hat{\vec{x}}_N \quad (5.142)$$

$$\begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} = \left. \frac{\partial^2 \mathcal{L}_k}{\partial \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix} \partial \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix}} \right|_{\vec{x}_k = \hat{\vec{x}}_k, \vec{u}_k = \hat{\vec{u}}_k} \quad (5.143)$$

and

$$\begin{bmatrix} \vec{q}_k \\ \vec{r}_k \end{bmatrix} = \left. \frac{\partial \mathcal{L}_k}{\partial \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix}} \right|_{\vec{x}_k = \hat{\vec{x}}_k, \vec{u}_k = \hat{\vec{u}}_k} - \begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} \begin{bmatrix} \hat{\vec{x}}_k \\ \hat{\vec{u}}_k \end{bmatrix} \quad (5.144)$$

where one must enforce $E > 0$, $Q_k \geq 0$ and $R_k > 0$ for all k .

In addition, for the I/ELQR approaches, assume one can linearize the forward dynamics with

$$F_k = \left. \frac{\partial f_k}{\partial \vec{x}_k} \right|_{\vec{x}_k = \hat{\vec{x}}_k, \vec{u}_k = \hat{\vec{u}}_k} \quad (5.145)$$

and

$$G_k = \left. \frac{\partial f_k}{\partial \vec{u}_k} \right|_{\vec{x}_k = \hat{\vec{x}}_k, \vec{u}_k = \hat{\vec{u}}_k} \quad (5.146)$$

with a linearization error of

$$\delta \vec{x}_{k+1} = \hat{\vec{x}}_{k+1} - F_k \hat{\vec{x}}_k - G_k \hat{\vec{u}}_k \quad (5.147)$$

This provides an affine forward dynamics model as

$$\hat{\vec{x}}_{k+1} = F_k \hat{\vec{x}}_k + G_k \hat{\vec{u}}_k + \delta \vec{x}_{k+1} \quad (5.148)$$

Lastly, for ELQR, assume one can linearize the backward dynamics with

$$\tilde{F}_k = \left. \frac{\partial \tilde{f}_k}{\partial \vec{x}_{k+1}} \right|_{\vec{x}_{k+1} = \hat{\vec{x}}_{k+1}, \vec{u}_k = \hat{\vec{u}}_k} \quad (5.149)$$

and

$$\tilde{G}_k = \left. \frac{\partial \tilde{f}_k}{\partial \vec{u}_k} \right|_{\vec{x}_k = \hat{\vec{x}}_k, \vec{u}_k = \hat{\vec{u}}_k} \quad (5.150)$$

with a linearization error of

$$\delta \tilde{\vec{x}}_k = \hat{\vec{x}}_k - \tilde{F}_k \hat{\vec{x}}_{k+1} - \tilde{G}_k \hat{\vec{u}}_k \quad (5.151)$$

This provides an affine backward dynamics model as

$$\hat{\vec{x}}_k = \tilde{F}_k \hat{\vec{x}}_{k+1} + \tilde{G}_k \hat{\vec{u}}_k + \delta \tilde{\vec{x}}_k \quad (5.152)$$

Backward LQR Synthesis via Cost-to-Go Functions

With this linearization-quadratization formulation in mind, consider the case where one has an affine forward dynamics model as

$$\vec{x}_{k+1} = F_k \vec{x}_k + G_k \vec{u}_k + \delta \vec{x}_{k+1} \quad (5.153)$$

and the cost function is quadratic, i.e.,

$$\mathcal{J} = \frac{1}{2} \vec{x}_N^T E \vec{x}_N + \vec{x}_N^T \vec{e} + \sum_{k=0}^{N-1} \frac{1}{2} \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix} + \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix}^T \begin{bmatrix} \vec{q}_k \\ \vec{r}_k \end{bmatrix} \quad (5.154)$$

Then the optimal solution, similar to the previous derivation of the standard LQR, can be synthesized by solving for the minimum cost-to-go function, \mathcal{V}_k , which solves the optimization problem

$$\mathcal{V}_k(\vec{x}_k) = \min_{\vec{u}_k[k] \text{ for } k=0, \dots, N-1} \mathcal{V}_{k+1}(\vec{x}_{k+1}) + \frac{1}{2} \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix}^T \begin{bmatrix} Q_k & S_k^T \\ S_k & R_k \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix} + \begin{bmatrix} \vec{x}_k \\ \vec{u}_k \end{bmatrix}^T \begin{bmatrix} \vec{q}_k \\ \vec{r}_k \end{bmatrix} \quad (5.155)$$

with final condition

$$\mathcal{V}_N(\vec{x}_N) = \frac{1}{2} \vec{x}_N^T E \vec{x}_N + \vec{x}_N^T \vec{e} \quad (5.156)$$

The solution to this general backwards LQR is obtained by assuming \mathcal{V}_k has the general quadratic form of

$$\mathcal{V}_k(\vec{x}_k) = \frac{1}{2} \vec{x}_k^T P_k \vec{x}_k + \vec{x}_k^T \vec{p}_k + c \quad (5.157)$$

where c is a constant. Then, the optimal $P_k > 0$ can be shown to be computed backwards via the Riccati difference equation

$$\begin{aligned} P_k = & F_k^T P_{k+1} F_k + Q_k \\ & - (G_k^T P_{k+1} F_k + S_k)^T \\ & (G_k^T P_{k+1} G_k + R_k)^{-1} \\ & (G_k^T P_{k+1} F_k + S_k) \end{aligned} \quad (5.158)$$

and the optimal \vec{p}_k is computed backwards via the recursive equation

$$\begin{aligned} \vec{p}_k = & F_k^T \vec{p}_{k+1} + F_k^T P_{k+1} \delta \vec{x}_k + \vec{q}_k \\ & - (G_k^T P_{k+1} F_k + S_k)^T (G_k^T P_{k+1} G_k + R_k)^{-1} \\ & (G_k^T \vec{p}_{k+1} + G_k^T P_{k+1} \delta \vec{x}_k + \vec{r}_k) \end{aligned} \quad (5.159)$$

with initial values, $P_N = E$, and $\vec{p}_N = \vec{e}$. Finally, the backward LQR optimal control law has the linear form

$$\begin{aligned} \vec{u}_k = \pi(\vec{x}_k) = & - (G_k^T P_{k+1} G_k + R_k)^{-1} \left((G_k^T P_{k+1} F_k + S_k) \vec{x}_k \right. \\ & \left. + (G_k^T \vec{p}_{k+1} + G_k^T P_{k+1} \delta \vec{x}_k + \vec{r}_k) \right) \end{aligned} \quad (5.160)$$

Iterative LQR Synthesis

The **iterative linear-quadratic regulator (ILQR)** solves the backwards LQR solution iteratively for the linearized dynamics and quadratized local cost function provided one has an initial nominal state trajectory, i.e., $(\hat{\vec{x}}_{k,0}, \hat{\vec{u}}_{k,0})$ for $k = 0, \dots, N-1$. If no initial nominal state trajectory is known, then one often sets $\hat{\vec{u}}_{k,0} = \vec{0}$ for $k = 0, \dots, N-1$ and solves for $\hat{\vec{x}}_{k,0}$ using the forward dynamics. The i^{th} iteration of the backwards LQR provides a new nominal locally optimal control law

$$\hat{\vec{u}}_{k,i} = \pi_{k,i}(\hat{\vec{x}}_{k,i}) \quad (5.161)$$

where one ends the iterative procedure for some convergence criteria, e.g.,

$$\sum_{k=0}^{N-1} \|\hat{\vec{u}}_{k,i} - \hat{\vec{u}}_{k,i-1}\|_2 < \tau \quad (5.162)$$

where τ is the convergence threshold. If the control has not converged, then one computes the $i+1^{\text{th}}$ iteration, by first computing the i^{th} nominal state trajectory for $k = 0, \dots, N-1$ forwards in time as

$$\hat{\vec{x}}_{k+1,i} = f_{k,i}(\hat{\vec{x}}_{k,i}, \pi_{k,i}(\hat{\vec{x}}_{k,i})) \quad (5.163)$$

Next, quadratizing and linearizing about $\hat{\vec{x}}_{k,i}$ and $\hat{\vec{u}}_{k,i}$ to obtain E_{i+1} , \vec{e}_{i+1} , $Q_{k,i+1}$, $R_{k,i+1}$, $S_{k,i+1}$, $\vec{q}_{k,i+1}$, $\vec{r}_{k,i+1}$, $F_{k,i+1}$, and $G_{k,i+1}$. Finally, resolving the backward LQR to obtain $\pi_{k,i+1}$ and checking for convergence.

Forward LQR Synthesis via Cost-to-Come Functions

With the backward linearization-quadratization formulation in mind, consider the case where one has an affine backward dynamics model as

$$\vec{x}_k = \tilde{F}_k \vec{x}_k + \tilde{G}_k \vec{u}_k + \delta \vec{x}_k \quad (5.164)$$

the cost function is the same as before. However, the optimal solution, similar to the previous derivation of the backward LQR, can be synthesized by assuming the minimum **cost-to-come function**, also known as the **forward value function**, $\tilde{\mathcal{V}}_k$, which an initial condition of

$$\mathcal{V}_0(\vec{x}_0) = \vec{0} \quad (5.165)$$

and a general quadratic form

$$\tilde{\mathcal{V}}_k(\vec{x}_k) = \frac{1}{2} \vec{x}_k^T \tilde{P}_k \vec{x}_k + \vec{x}_k^T \tilde{p}_k + \tilde{c} \quad (5.166)$$

where \tilde{c} is a constant.

Then, the optimal $\tilde{P}_k > 0$ is computed forwards via the Riccati difference equation

$$\begin{aligned} \tilde{P}_{k+1} = & \tilde{F}_k^T (\tilde{P}_k + Q_k) \tilde{F}_k \\ & - \left(\tilde{G}_k^T (\tilde{P}_k + Q_k) \tilde{F}_k + S_k \tilde{F}_k \right)^T \\ & \left(\tilde{G}_k^T (\tilde{P}_k + Q_k) \tilde{G}_k + R_k + S_k \tilde{G}_k + \tilde{G}_k^T S_k^T \right)^{-1} \\ & \left(\tilde{G}_k^T (\tilde{P}_k + Q_k) \tilde{F}_k + S_k \tilde{F}_k \right) \end{aligned} \quad (5.167)$$

and \tilde{p}_k is computed forwards via the recursive equation

$$\begin{aligned} \tilde{p}_{k+1} = & \tilde{F}_k^T (\tilde{p}_k + \vec{q}_k) + \tilde{F}_k^T (\tilde{P}_k + Q_k) \delta \tilde{x}_k \\ & \left(\tilde{G}_k^T (\tilde{P}_k + Q_k) \tilde{F}_k + S_k \tilde{F}_k \right)^T \\ & \left(\tilde{G}_k^T (\tilde{P}_k + Q_k) \tilde{G}_k + R_k + S_k \tilde{G}_k + \tilde{G}_k^T S_k^T \right)^{-1} \\ & \left(\tilde{G}_k^T (\tilde{p}_k + \vec{q}_k) + (\tilde{G}_k^T (\tilde{P}_k + Q_k) + S_k) \delta \tilde{x}_k + \vec{r}_k \right) \end{aligned} \quad (5.168)$$

with initial values, $\tilde{P}_0 = [0]$, and $\tilde{p}_0 = \vec{0}$.

Finally, the forward LQR optimal control law has a linear form as

$$\begin{aligned} \vec{u}_k = \tilde{\pi}(\vec{x}_{k+1}) = & - \left(\tilde{G}_k^T (\tilde{P}_k + Q_k) \tilde{G}_k + R_k + S_k \tilde{G}_k + \tilde{G}_k^T S_k^T \right)^{-1} \\ & \left(\left(\tilde{G}_k^T (\tilde{P}_k + Q_k) \tilde{F}_k + S_k \tilde{F}_k \right) \vec{x}_{k+1} \right. \\ & \left. + \left(\tilde{G}_k^T (\tilde{P}_k + Q_k) + (\tilde{G}_k^T (\tilde{P}_k + Q_k) + S_k) \delta \tilde{x}_k + \tilde{r}_k \right) \right) \end{aligned} \quad (5.169)$$

Extended LQR Synthesis

The **extended linear-quadratic regulator (ELQR)** which, like the ILQR, linearizes the forward dynamics and quadratizes the local cost function in the backward iteration of the cost-to-go functions, but, instead of recomputing the forward dynamics, the ELQR linearizes the backward dynamics, quadratizes the local cost function, and computes the forward LQR in the forwards iteration of the cost-to-come functions. This requires twice as many linearization and quadratization computations than the ILQR and twice as much memory for the dynamics for forward and backward, but can provide significantly less iterations than the ILQR for certain types of nonlinear and non-quadratic OCPs.

To optimally combine the two procedures of a backward and forward LQR for an accurate linearization/quadratization point, the Extended LQR takes advantage of the minimum-cost sequence of states and controls at time step k as the total-cost, $\mathcal{V}_k^{tot}(\vec{x}_k)$, defined as the sum of the minimum cost-to-go and cost-to-come functions, i.e.,

$$\mathcal{V}_k^{tot}(\vec{x}_k) = \mathcal{V}_k(\vec{x}_k) + \tilde{\mathcal{V}}_k(\vec{x}_k) = \frac{1}{2} \vec{x}_k^T (P_k + \tilde{P}_k) \vec{x}_k + \vec{x}_k^T (\tilde{P}_k + \tilde{P}_k) + (c + \tilde{c}) \quad (5.170)$$

Then, the state trajectory for $k = 1, \dots, N$ for which the $\mathcal{V}_k^{tot}(\vec{x}_k)$ is minimum-cost can be written as simply

$$\hat{\vec{x}}_k = \underset{\vec{x}_k}{\operatorname{argmin}} \mathcal{V}_k^{tot}(\vec{x}_k) = -(P_k + \tilde{P}_k)^{-1} (\tilde{P}_k + \tilde{P}_k) \quad (5.171)$$

Then, the associated forward and backward control laws are given by

$$\hat{\vec{x}}_{k+1} = f_k(\hat{\vec{x}}_k, \pi_k(\hat{\vec{x}}_k)) \quad (5.172)$$

and

$$\hat{\vec{x}}_k = \tilde{f}_k(\hat{\vec{x}}_{k+1}, \tilde{\pi}_k(\hat{\vec{x}}_{k+1})) \quad (5.173)$$

where $\hat{\vec{u}}_k = \pi_k(\hat{\vec{x}}_k) = \tilde{\pi}_k(\hat{\vec{x}}_{k+1})$. This can be understood as the LQR-dual of the fixed-interval Kalman smoother (FI-KS) discussed later in this textbook on state estimation, or as **LQR-smoothing**.

Each iteration of the ELQR is performed as follows. First, one computes the backward nominal control input using the previous forward LQR control law as

$$\hat{\vec{u}}_{k,i-} = \tilde{\pi}_{k,i-1}(\hat{\vec{x}}_{k+1,i-}) \quad (5.174)$$

simultaneously with the backward nominal state trajectory using the backwards dynamics as

$$\hat{\vec{x}}_{k,i-} = \tilde{f}_{k,i}(\hat{\vec{x}}_{k+1,i-}, \hat{\vec{u}}_{k,i-}) \quad (5.175)$$

which can be initialized at $i = 1$ according to some prior information or by assuming $\tilde{\pi}_{k,0} = \vec{0}$ for all k , and $\hat{x}_{N,0\leftarrow} = 0$. Next, one linearizes the forward dynamics and quadratizes the local cost function about $\hat{x}_{k,i\leftarrow}$ and $\hat{u}_{k,i\leftarrow}$. Then, one computes the backward LQR control law as $\pi_{k,i}(\vec{x}_k)$.

This backward LQR control law provides the forward control input as

$$\hat{u}_{k,i\rightarrow} = \pi_{k,i}(\hat{x}_{k,i\rightarrow}) \quad (5.176)$$

simultaneously with the forward nominal state trajectory using the forward dynamics as

$$\hat{x}_{k+1,i\rightarrow} = f_{k,i}(\hat{x}_{k,i\rightarrow}) \quad (5.177)$$

Next, one linearizes the backwards dynamics and quadratizes the local cost function about $\hat{x}_{k,i\rightarrow}$ and $\hat{u}_{k,i\rightarrow}$. Then, one computes the forward LQR control as $\tilde{\pi}_{k,i}(\vec{x}_{k+1})$ up until $k = N - 1$ with $\tilde{P}_0 = [0]$ and $\tilde{p} = \vec{0}$. In addition, the initial quadratization point for time step k for the next backward LQR pass is calculated using the minimum-cost state formula, i.e.,

$$\hat{x}_N = -(P_N + \tilde{P}_N)^{-1}(\vec{p}_N + \tilde{p}_N) \quad (5.178)$$

Finally, one can check for convergence at either pass based on π and $\tilde{\pi}$

References

For more information, please refer to the following

- Li, W. and Todorov, E., “Iterative Linear Quadratic Regulator Design for Nonlinear Biological Movement Systems,” in *Proceedings of the 1st International Conference on Informatics in Control, Automation and Robotics*, Setubal, Portugal, 2004
- Van Den Burg, J., “Extended LQR: Locally-Optimal Feedback Control for Systems with Non-Linear Dynamics and Non-Quadratic Cost,” in *Inaba, M., Corke, P. (eds) Robotics Research: Springer Tracts in Advanced Robotics*, Vol. 114. 2016

5.5 Receding-Horizon Linear-Quadratic Regulator

Another sub-optimal approach to solving discrete-time OCPs is **receding horizon control (RHC)**, also known as **model predictive control (MPC)**, which optimizes the control input over a **control horizon** of length $M < N$ and re-optimizes at *every* time step of the control process, k , instead of optimizing over the entire time horizon N , also known as the **prediction horizon** in RHC. This approach also is more conducive for constrained OCPs as it can enforce state and input constraints remain true over the prediction horizon. Thus, the name RHC derives from the control and prediction horizons receding with each time step while the name MPC derives from the optimization method using a dynamics model to predict the state forward in time for the OCP solved at every time step.

The general receding horizon OCP can be constructed as

$$\begin{aligned}
 \vec{u}[0]^{opt}, \dots, \vec{u}[M-1]^{opt} &= \underset{\vec{u}}{\operatorname{argmin}} \quad J = \mathcal{E}(\vec{x}[M]) + \sum_{k=0}^{M-1} \mathcal{L}(\vec{x}[k], \vec{u}[k]) \\
 \text{subject to: } \vec{x}[k+1] &= f(\vec{x}[k], \vec{u}[k]) \\
 \text{initial condition: } \vec{x}_0 & \\
 \text{state constraints: } \vec{x}[k] &\in X \quad \forall 0 \leq k \leq N \\
 \text{input constraints: } \vec{u}[k] &\in \mathcal{U} \quad \forall 0 \leq k \leq M
 \end{aligned} \tag{5.179}$$

It is important to note that the state constraints are enforced over the prediction horizon and the input constraints are enforced over the control horizon. In addition, for $M \leq k \leq N$, the control input $u[k]$, must be specified. RHC often is computed in real-time and cannot be precomputed as a known controller. Thus, RHC serves as an open-loop control scheme since it recomputes its action at every time step and does not explicitly provide a control law for the feedback loop, though it does reuse the “new” state as the “new” initial in its re-optimization of the problem every time step. This feature of RHC may be especially desirable if the real process dynamics are not exactly represented in the dynamics model, especially nonlinearities. Thus, RHC continues to be used in many circumstances.

When the receding horizon OCP has linear dynamics, a quadratic cost, and minimum and/or maximum input and state constraints, one has a constrained receding horizon LQ OCP which can be stated as

$$\begin{aligned}
 \vec{u}_0^{opt}, \dots, \vec{u}_{M-1}^{opt} &= \underset{\vec{u}}{\operatorname{argmin}} \quad J = \vec{x}^T[M] E \vec{x}[M] \\
 &\quad + \sum_{k=0}^{M-1} \vec{x}^T[k] Q \vec{x}[k] \\
 &\quad + \vec{u}^T[k] R \vec{u}[k] \\
 \text{subject to: } \vec{x}[k+1] &= F \vec{x}[k] + G \vec{u}[k] \\
 \text{initial condition: } \vec{x}_0 & \\
 \text{state constraints: } \vec{x}_{min} &\leq \vec{x}[k] \leq \vec{x}_{max} \quad \forall 0 \leq k \leq N \\
 \text{input constraints: } \vec{u}_{min} &\leq \vec{u}[k] \leq \vec{u}_{max} \quad \forall 0 \leq k \leq M
 \end{aligned} \tag{5.180}$$

for which the optimal control sequence is the receding horizon linear-quadratic regulator (RHLQR)

To formulate the RHLQR solution for an LTI state-space system with a quadratic cost function with constraints, one typically uses quadratic programming on the entire control horizon. Thus, based on the discrete-time linear dynamics and initial condition, one can write out the general solution of the state as a “large” linear function, i.e.

$$\begin{bmatrix} \vec{x}[0] \\ \vdots \\ \vec{x}[M] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ G & 0 & 0 & \cdots & 0 \\ FG & G & 0 & \cdots & 0 \\ F^2G & FG & G & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F^{M-1}G & F^{M-2}G & F^{M-3}G & \cdots & G \end{bmatrix} \begin{bmatrix} \vec{u}[0] \\ \vdots \\ \vec{u}[M-1] \end{bmatrix} + \begin{bmatrix} I \\ F \\ \vdots \\ F^M \end{bmatrix} \vec{x}_0 \tag{5.181}$$

which one can rewrite succinctly by defining the above vectors and matrices as

$$\vec{x} = \mathbf{G}\vec{u} + \mathbf{F}\vec{x}_0 = \mathbf{G}\vec{u} + \vec{x}_0 \quad (5.182)$$

Then, using these new terms and the following block diagonal matrices

$$\mathbf{Q} = \begin{bmatrix} Q & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & Q & 0 \\ 0 & \cdots & 0 & E \end{bmatrix} \quad (5.183)$$

and

$$\mathbf{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix} \quad (5.184)$$

the cost function for the discrete-time LQ OCP can be rewritten using vector notation as

$$J = \vec{x}^T \mathbf{Q} \vec{x} + \vec{u}^T \mathbf{R} \vec{u} \quad (5.185)$$

and, by substitution for \vec{x} , one has

$$J = (\mathbf{G}\vec{u} + \vec{x}_0)^T \mathbf{Q} (\mathbf{G}\vec{u} + \vec{x}_0) + \vec{u}^T \mathbf{R} \vec{u} \quad (5.186)$$

$$J = \vec{x}_0^T \mathbf{Q} \vec{x}_0 + 2\vec{x}_0^T \mathbf{Q} \mathbf{G} \vec{v} + \vec{v}^T \mathbf{G}^T \mathbf{Q} \mathbf{G} \vec{v} + \vec{v}^T \mathbf{R} \vec{v} \quad (5.187)$$

Finally, noting that the first term can be ignored since it is not affected by the choice of the argument \vec{v} , one can assign

$$\tilde{Q} = \mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R} \quad (5.188)$$

and

$$c^T = \vec{x}_0^T \mathbf{Q} \mathbf{G} \quad (5.189)$$

to rewrite the cost function in the form of the QP cost function.

$$J = \frac{1}{2} \vec{u}^T \tilde{Q} \vec{u} + c^T \vec{u} \quad (5.190)$$

By inspection, one can see that the only parameter that changes in this optimization from one time step to the next is the term \vec{x}_0 which changes with \vec{x}_0 which is the current state at each subsequent time step as one moves forward in time.

In order to put the inequality constraints into the QP form above, stacking techniques can be used as follows. For the input constraints,

$$\begin{bmatrix} \vec{u}_{min} \\ \vdots \\ \vec{u}_{min} \end{bmatrix} \leq I \vec{u} \leq \begin{bmatrix} \vec{u}_{max} \\ \vdots \\ \vec{u}_{max} \end{bmatrix} \quad (5.191)$$

for the state constraints for $1 \leq k \leq M$,

$$\begin{bmatrix} \vec{x}_{min} \\ \vdots \\ \vec{x}_{min} \end{bmatrix} - \vec{x}_0 \leq \mathbf{G} \vec{u} \leq \begin{bmatrix} \vec{x}_{max} \\ \vdots \\ \vec{x}_{max} \end{bmatrix} - \vec{x}_0 \quad (5.192)$$

For the state constraints for $M+1 \leq k \leq N$,

$$\begin{bmatrix} \vec{x}_{min} \\ \vdots \\ \vec{x}_{min} \end{bmatrix} \leq \begin{bmatrix} \vec{x}[M+1] \\ \vdots \\ \vec{x}[N] \end{bmatrix} \leq \begin{bmatrix} \vec{x}_{max} \\ \vdots \\ \vec{x}_{max} \end{bmatrix} \quad (5.193)$$

where the stacked states can be written as

$$\begin{bmatrix} \vec{x}[M+1] \\ \vdots \\ \vec{x}[N] \end{bmatrix} = \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} \vec{x}_M \quad (5.194)$$

and using the solution for the state at time step M

$$\vec{x}_M = F^M \vec{x}_0 + [F^{M-1} G \ \cdots \ G] \vec{v} \quad (5.195)$$

one can write

$$\begin{bmatrix} \vec{x}[M+1] \\ \vdots \\ \vec{x}[N] \end{bmatrix} = \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} \left(F^M \vec{x}_0 + [F^{M-1} G \ \cdots \ G] \vec{u} \right) \quad (5.196)$$

Then, after substitution and rearrangement, the state constraints for $M+1 \leq k \leq N$ can finally be put into the QP form as

$$\begin{bmatrix} \vec{x}_{min} \\ \vdots \\ \vec{x}_{min} \end{bmatrix} - \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} F^M \vec{x}_0 \leq \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} [F^{M-1} G \ \cdots \ G] \vec{u} \leq \begin{bmatrix} \vec{x}_{max} \\ \vdots \\ \vec{x}_{max} \end{bmatrix} - \begin{bmatrix} F \\ \vdots \\ F^{N-M} \end{bmatrix} F^M \vec{x}_0 \quad (5.197)$$

It should also be noted that rate constraints on the inputs can also be incorporated into the QP problem form in a similar fashion, the minimum and maximum values for \vec{x} and \vec{u} can be dependent on k , and equality constraints can be represented by choosing the element of \vec{x}_{min} equal to the same element in \vec{x}_{max} for any particular k .

Linear-in-Control Systems and Adaptive Control Theory

6.1 Linear-in-Control Systems and Dynamic Inversion

Linear-In-Control Dynamical System

Furthermore, flight vehicles are not inherently LTI systems and therefore one often linearizes the nonlinear, time-invariant dynamics over a grid of flight conditions in order to design the MIMO LTI state feedback controllers at those conditions and interpolate control gains, an adaptive technique known as gain-scheduling. This technique necessitates smooth transitions between flight conditions such that the linearized model is a good approximation to the dynamics and robustness to the neglected higher-order terms (HOT) in the dynamics about those flight conditions. However, during some flight maneuvers, one may not be able to sufficiently neglect the nonlinear dynamics for the flight vehicles. An alternative approach to gain-scheduling adaptive control is to use **dynamic inversion (DI) control** which transforms a nonlinear, time-invariant dynamical system into an LTI system via inversion which can then be controlled using a MIMO LTI controller.

Dynamic inversion control will assume a square system, i.e. $n_u = n_y$, \vec{x} is available for feedback, then a **linear-in-control dynamical system**, also known as a **control-affine dynamical system**

$$\begin{aligned}\dot{\vec{x}} &= f(\vec{x}) + g(\vec{x})\vec{u} \\ \vec{y} &= h(\vec{x})\end{aligned}\tag{6.1}$$

which is suitable for many flight vehicle dynamics, and that one desires to control the output $\vec{y}(t)$ to track a commanded reference signal, $\vec{r}(t)$. Thus, $\vec{y}(t)$ is also known as the **controlled variable** in this context and the tracking error, \vec{e} , is defined as

$$\vec{e}(t) = \vec{r}(t) - \vec{y}(t)\tag{6.2}$$

This type of dynamic inversion differentiates the controlled variable, $\vec{y}(t)$, until the control, $\vec{u}(t)$, appears in the expression for the derivative. Thus, this type of dynamic inversion control is also known as **input-output feedback linearization control** and will be presented here first for linear, time-invariant

systems as an introduction to the concept and linear-in-control systems afterwards where it shows its true usefulness.

Lastly, it is important to note that this method relies heavily on the differentiation of the dynamical system which may not be suitably robust to model uncertainties. To overcome this result, nonlinear dynamic inversion is often used alongside adaptive control methods for linear-in-control systems to account for **matched uncertainties** in the dynamics model, i.e. those that can be canceled by choosing some $\vec{u}(t)$, as opposed to **unmatched uncertainties**, i.e. those that can't be canceled.

The following subsections will introduce these basic control designs which can be extended to be robust to unmatched uncertainties. An exhaustive discussion of these control methods including stability and robustness is beyond the scope of this textbook and are typically addressed in nonlinear systems and/or adaptive control courses.

LTI Dynamic Inversion Control

As an introduction to dynamic inversion control, consider the LTI system

$$\begin{aligned}\dot{\vec{x}} &= A\vec{x} + B\vec{u} \\ \vec{y} &= C\vec{x}\end{aligned}\tag{6.3}$$

where the derivative of \vec{y} can be shown to be

$$\dot{\vec{y}} = C\dot{\vec{x}} = CA\vec{x} + CB\vec{u}\tag{6.4}$$

Then, one can define the inner **feedback linearization loop** as the control law

$$\vec{u} = (CB)^{-1}(-CA\vec{x} + \dot{\vec{r}} + \vec{v})\tag{6.5}$$

where \vec{v} is the **virtual control input**.

Substituting this control law for \vec{y} , one has

$$\dot{\vec{y}} = CA\vec{x} + CB(CB)^{-1}(-CA\vec{x} + \dot{\vec{r}} + \vec{v})\tag{6.6}$$

$$\dot{\vec{y}} = CA\vec{x} - CA\vec{x} + \dot{\vec{r}} + \vec{v}\tag{6.7}$$

$$\vec{v} = \dot{\vec{y}} - \dot{\vec{r}}\tag{6.8}$$

or as the **error dynamics**

$$\dot{\vec{e}} = -\vec{v}\tag{6.9}$$

which has n_y poles at $s = 0$. Thus, selecting \vec{u} in this way allowed one to cancel the $CA\vec{x}$ term and relate it to the tracking error without the CB term, making the system from \vec{v} to \vec{y} appear like a linear system with poles at the origin.

Then, one can design the **outer tracking loop** as a state feedback control law, K , using MIMO LTI control on the tracking error, \vec{e} , i.e.

$$\vec{v} = K\vec{e}\tag{6.10}$$

which provides error dynamics

$$\dot{\vec{e}} = -K\vec{e}\tag{6.11}$$

and is stable if and only if $K > 0$, i.e. positive definite, and is typically a diagonal matrix to keep the control channels in the outer-loop decoupled. Thus, the overall **LTI dynamic inversion (LDI) controller** is given by

$$\vec{u} = (CB)^{-1} (-CA\vec{x} + \dot{\vec{r}} + K\vec{e}) \quad (6.12)$$

which sums the feedback linearization loop output, the LTI control gain on the tracking error, and a feedforward term on the reference derivative $\dot{\vec{r}}$.

Furthermore, substituting the LDI controller into the state equation, one has closed-loop system dynamics as

$$\begin{aligned} \dot{\vec{x}} &= A\vec{x} + B(CB)^{-1}(-CA\vec{x} + \dot{\vec{r}} + \vec{v}) \\ \dot{\vec{x}} &= (I - B(CB)^{-1})A\vec{x} + B(CB)^{-1}(\dot{\vec{r}} + \vec{v}) \end{aligned} \quad (6.13)$$

To analyze the stability of this system, one must not only understand the error dynamics, i.e. $\dot{\vec{e}}$, but also the **zero dynamics**, i.e. $\vec{y}(t) = 0$ or $\vec{v} = -\dot{\vec{r}}$ such that

$$\dot{\vec{x}} = (I - B(CB)^{-1})A\vec{x} = A_z\vec{x} \quad (6.14)$$

Note that the dimension of the \vec{e} is $n_y < n_x$, thus, the remaining $n_x - n_y$ system poles must be LHP stable for the dynamic inversion controller to be stable. These poles are unobservable by selecting the controlled variable $\vec{y} = C\vec{x}$ as the term $I - B(CB)^{-1}$ can be shown as the projection on the null space of C along the range of B . Thus, $(I - B(CB)^{-1})A = A_z$ describes the dynamics in the null space of C and in range perpendicular of B . This demonstrates that designing C , i.e. the controlled variables of the state, is crucial to the stability of LTI dynamic inversion control.

Lastly, it should be noted that this design assumed CB was invertible and non-zero. However, one can note that the n^{th} derivative of \vec{y} as

$$\vec{y}^{[n]} = CA^{n+1}\vec{x} + C[A^nB \cdots AB \ B] \begin{bmatrix} \vec{u} \\ \vdots \\ \vec{u}^{[n-1]} \\ \vec{u}^{[n]} \end{bmatrix} \quad (6.15)$$

Thus, if CB is zero, one can simply take the smallest n^{th} derivative which has a non-zero CA^nB term, i.e.

$$\vec{y}^{[n]} = CA^{n+1}\vec{x} + CA^nB\vec{u} \quad (6.16)$$

where one can form the dynamic inversion controller as

$$\vec{u} = (CA^{n-1}B)^{-1}(-CA^n\vec{x} + \vec{r}^{[n]} + \vec{v}) \quad (6.17)$$

which provides

$$\vec{e}^{[n]} = -\vec{v} \quad (6.18)$$

which has $n_y \times n$ poles at the origin.

However, if CB is non-zero and singular, the system must be controllable and taking $n = n_x$, one can form the controllability matrix as

$$C = [A^{n_x} B \cdots AB B] \quad (6.19)$$

which has rank n_x . However, the C matrix has rank n_u , thus one cannot use a strict inverse transformation. However, defining the **pseudoinverse** of CC as

$$(CC)^+ = (CC)^T \left((CC)(CC)^T \right)^{-1} \quad (6.20)$$

one can form the dynamic inversion controller as

$$\begin{bmatrix} \vec{u} \\ \vdots \\ \vec{u}^{[n_x-1]} \\ \vec{u}^{[n_x]} \end{bmatrix} = (CC)^+ \left(-CA^{n_x} \vec{x} + \vec{r}^{[n_x]} + \vec{v} \right) \quad (6.21)$$

which provides

$$\vec{e}^{[n_x]} = -\vec{v} \quad (6.22)$$

which has $n_y \times n_x$ poles at the origin due to the pseudoinverse.

In both cases, one can design the **outer tracking loop** as a state feedback control law, K , using MIMO LTI control on the tracking error, \vec{e} , i.e.

$$\vec{v} = K_{n-1} \vec{e}^{[n-1]} + \cdots + K_0 \vec{e} \quad (6.23)$$

which provides error dynamics

$$\dot{\vec{e}} + K_{n-1} \vec{e}^{[n-1]} + \cdots + K_0 \vec{e} = 0 \quad (6.24)$$

where the gain matrices, K_i for $i = 0, \dots, n_x$ are selected to make the error dynamics stable. Notably, these cases require one to compute the tracking error and its n derivatives. Furthermore, for CB as non-zero and singular, one must also use a dynamical system to extract $\vec{u}(t)$ from its derivative vector $[\vec{u} \ \cdots \ \vec{u}^{[n_x-1]} \ \vec{u}^{[n_x]}]^T$.

Nonlinear Dynamic Inversion Control

Next, consider the linear-in-control, time-invariant system

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}) + g(\vec{x}) \vec{u} \\ \vec{y} &= h(\vec{x}) \end{aligned} \quad (6.25)$$

where the derivative of \vec{y} can be shown to be

$$\dot{\vec{y}} = \frac{\partial h}{\partial \vec{x}} \dot{\vec{x}} = \frac{\partial h}{\partial \vec{x}} f(\vec{x}) + \frac{\partial h}{\partial \vec{x}} g(\vec{x}) \vec{u} \quad (6.26)$$

where $\frac{\partial h}{\partial \vec{x}} f(\vec{x})$ is known as the **Lie derivative** of $h(\vec{x})$ along $f(\vec{x})$. Then, one can define the inner **feedback linearization loop**

$$\vec{u} = \left(\frac{\partial h}{\partial \vec{x}} g(\vec{x}) \right)^{-1} \left(-\frac{\partial h}{\partial \vec{x}} f(\vec{x}) + \dot{\vec{r}} + \vec{v} \right) \quad (6.27)$$

with **virtual control input**, $\vec{v}(t)$.

Substituting this control law for $\dot{\vec{y}}$, one obtains the **error dynamics**

$$\dot{\vec{e}} = -\vec{v} \quad (6.28)$$

Then, one can design the **outer tracking loop** as a state feedback control law, K , using MIMO LTI control on the tracking error, \vec{e} , i.e.

$$\vec{v} = K\vec{e} \quad (6.29)$$

which provides error dynamics

$$\dot{\vec{e}} = -K\vec{e} \quad (6.30)$$

and is stable if and only if $K > 0$, i.e. positive definite, and is typically a diagonal matrix to keep the control channels in the outer-loop decoupled.

Thus, the overall **nonlinear dynamic inversion (NDI) controller** is given by

$$\vec{u} = \left(\frac{\partial h}{\partial \vec{x}} g(\vec{x}) \right)^{-1} \left(-\frac{\partial h}{\partial \vec{x}} f(\vec{x}) + \dot{\vec{r}} + K\vec{e} \right) \quad (6.31)$$

which sums the feedback linearization loop output, the LTI control gain on the tracking error, and a feedforward term on the reference derivative $\dot{\vec{r}}$. Notably \vec{u} requires one to know the nonlinear dynamics or to be able to use lookup tables to compute $\frac{\partial h}{\partial \vec{x}} f(\vec{x})$ and $\frac{\partial h}{\partial \vec{x}} g(\vec{x})$. Furthermore, similar to the case where CB is singular or zero for LDI, if $\frac{\partial h}{\partial \vec{x}} g(\vec{x})$ is singular, one must use successive Lie derivatives to form suitable feedback linearization controllers.

Furthermore, substituting the NDI controller into the state equation, one has closed-loop system dynamics as

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}) + g(\vec{x}) \left(\frac{\partial h}{\partial \vec{x}} g(\vec{x}) \right)^{-1} \left(-\frac{\partial h}{\partial \vec{x}} f(\vec{x}) + \dot{\vec{r}} + K\vec{e} \right) \\ \dot{\vec{x}} &= (I - g(\vec{x})) \left(\frac{\partial h}{\partial \vec{x}} g(\vec{x}) \right)^{-1} \left(-\frac{\partial h}{\partial \vec{x}} \right) f(\vec{x}) + g(\vec{x}) \left(\frac{\partial h}{\partial \vec{x}} g(\vec{x}) \right)^{-1} (\dot{\vec{r}} + \vec{v}) \end{aligned} \quad (6.32)$$

with **zero dynamics**, i.e. $\vec{y}(t) = 0$ or $\vec{v} = -\dot{\vec{r}}$ such that

$$\dot{\vec{x}} = (I - g(\vec{x})) \left(\frac{\partial h}{\partial \vec{x}} g(\vec{x}) \right)^{-1} \left(-\frac{\partial h}{\partial \vec{x}} \right) f(\vec{x}) \quad (6.33)$$

which can be linearized at specific flight conditions to check the suitability of the controlled variable, $\vec{y} = h(\vec{x})$, or can be simulated at different initial conditions and checked.

Incremental Nonlinear Dynamic Inversion

To improve the robustness to model uncertainty in the NDI control strategy, one can alternatively design an **incremental nonlinear dynamic inversion (INDI) controller** instead of a full NDI which calculates the required *change* to the control input as opposed to the full control input. Thus, a full system model is not required, only a local part of the model, making INDI more robust.

Adaptive Control with Nonlinear Dynamic Inversion

Another control strategy to improve the performance of NDI to model uncertainty is to use **NDI-Based adaptive control**. Here, the reference signal, \vec{r} , and its derivative, $\dot{\vec{r}}$, are defined via a specified dynamical system called the **reference model** which is modeled to track the plant state, i.e. regulate the tracking error $\vec{e} = \vec{r} - \vec{x}$ to zero, and the control system incorporates an additional adaptive feedback loop to account for the matched uncertainties in the dynamics.

As an example of NDI-based adaptive control, consider a linear-in-control dynamical system with additive uncertainty $\Delta(\vec{x})$, one has

$$\dot{\vec{x}} = f(\vec{x}) + g(\vec{x})\vec{u} + \Delta(\vec{x}, \vec{u}) \quad (6.34)$$

where one forms the NDI-based adaptive controller as

$$\vec{u} = g(\vec{x})^{-1} \left(-f(\vec{x}) + \dot{\vec{r}} + K\vec{e} - \hat{\Delta}(\vec{x}, \vec{u}, \vec{\theta}) \right) \quad (6.35)$$

where $\hat{\Delta}$ is the new adaptive control signal that depends on \vec{x} , \vec{u} , and internal design parameters, $\vec{\theta}$, and is nominally used to cancel the uncertainty $\Delta(\vec{x})$. Thus, in this case, the error dynamics for the model reference are

$$\dot{\vec{e}} = \dot{\vec{r}} - \dot{\vec{x}} \quad (6.36)$$

$$\dot{\vec{e}} = \dot{\vec{r}} - f(\vec{x}) - g(\vec{x})\vec{u} - \Delta(\vec{x}, \vec{u}) \quad (6.37)$$

$$\dot{\vec{e}} = \dot{\vec{r}} - f(\vec{x}) - \left(-f(\vec{x}) + \dot{\vec{r}} + K\vec{e} - \hat{\Delta}(\vec{x}, \vec{u}, \vec{\theta}) \right) - \Delta(\vec{x}) \quad (6.38)$$

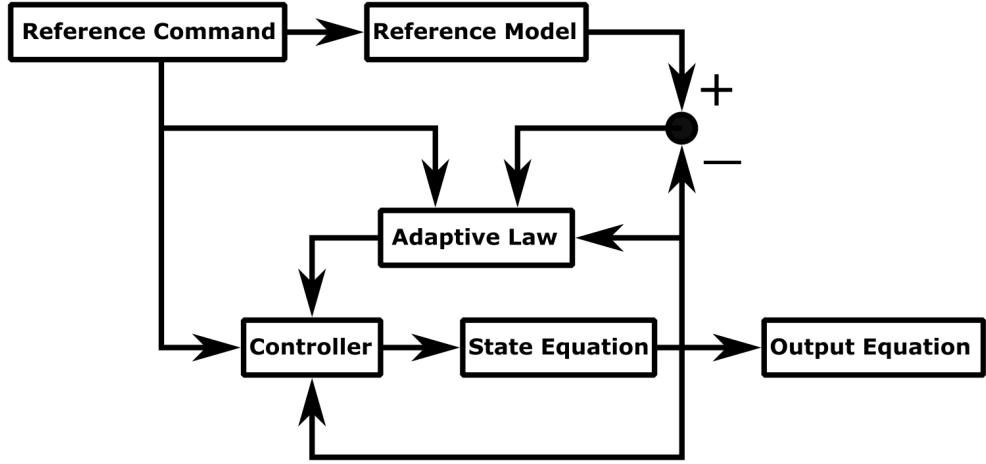
$$\dot{\vec{e}} = -K\vec{e} + \hat{\Delta}(\vec{x}, \vec{u}, \vec{\theta}) - \Delta(\vec{x}, \vec{u}) \quad (6.39)$$

where one ideally desires to design the adaptive element $\hat{\Delta}(\vec{x}, \vec{u}, \vec{\theta})$ to cancel the uncertainty, $\Delta(\vec{x})$, and design K to stabilize the n_x system poles. However, the stability and performance of such a system is dependent on the nature of $\Delta(\vec{x}, \vec{u})$ and the state trajectory $\vec{x}(t)$ where often $\hat{\Delta}(\vec{x}, \vec{u}, \vec{\theta})$ may not even converge to $\Delta(\vec{x}, \vec{u})$, but the system is still stable. Furthermore, adaptive neural networks have been used to perform the uncertainty cancellation term $\hat{\Delta}(\vec{x}, \vec{u})$ and adaptive control can be performed with and without NDI. In the following sections, adaptive control will be discussed further.

6.2 Gain-Scheduled Adaptive Control

6.3 Model Reference Adaptive Control

A generic block diagram for a system operating with Model Reference Adaptive Control (MRAC) is



In essence, an MRAC system consists of a controller whose gains are updated online using an adaptive law. The adaptive law is a function of the system output, the reference command, and the tracking error, i.e. the difference between the system output and the reference model output which specifies the desired trajectories for the system to follow. Finally, the controller computes its commands based on the reference command, the system output, and the online adjusted parameters from the adaptive law. Per design, the adaptive controller forces the system output to follow the desired reference commands while operating in the presence of system parameter uncertainties. The main objective of the controller is to maintain consistent performance of the closed-loop system in the presence of uncertainties and unknown variations in the system parameters. When the true system parameters are unknown, one may attempt to estimate control gains online using available measurements. This approach is referred to as **direct adaptive control**. Alternatively, the gains can be estimated online by solving system design equations that relate the uncertain parameters to the measured signals in the system. This is referred to as **indirect adaptive control**. MRAC systems can be designed using either approach as well as combining both approaches as **hybrid adaptive control**.

This chapter will consider command tracking algorithms for continuous-time dynamical systems, i.e.

$$\begin{aligned}\dot{\vec{x}} &= f(t, \vec{x}, \vec{u}, \vec{\theta}, \vec{\eta}) \\ \vec{y} &= h(t, \vec{x}, \vec{u}, \vec{\theta}, \vec{\eta})\end{aligned}\tag{6.40}$$

with constant vector of uncertain parameters, $\vec{\theta}$, bounded environmental disturbances, $\vec{\eta}$, and \vec{x} is available for full state feedback control, i.e. $\vec{y}(t) = \vec{x}(t)$. Then, the command tracking control problem involves designing the control input \vec{u} so that the regulated output $\vec{y}(t)$ tracks a given bounded reference signal $\vec{r}(t) \in \mathbb{R}^n$ in the presence of system uncertainties, $\vec{\theta}$ and environmental disturbances, $\vec{\eta}(t)$. Specifically, one must find $\vec{u}(t)$ such that the command tracking error

$$\vec{e} = \vec{y}(t) - \vec{r}(t)\tag{6.41}$$

becomes sufficiently small as $t \rightarrow \infty$ and all the signals in the corresponding closed-loop system remain uniformly bounded in time. Note that if one could achieve $\vec{e}(t) \rightarrow 0$ as $t \rightarrow \infty$, then asymptotic command

tracking would be achieved. However, this may not be feasible, in which case, the control objective would be to achieve uniform ultimate boundedness of the command tracking error, i.e.

$$\|\vec{e}\| \leq \epsilon, \quad \forall t \geq T \quad (6.42)$$

where $\epsilon > 0$ is the desired tracking tolerance and T is some finite time.

Direct MRAC

This subsection will consider the extension of the direct MRAC design to a **linear-in-control dynamical system**

$$\dot{\vec{x}} = A\vec{x} + B\Lambda(\vec{u} + f(\vec{x})) \quad (6.43)$$

where $\vec{x} \in \mathbb{R}^n$ is the system state, $\vec{u} \in \mathbb{R}^p$ is the control input, $B \in \mathbb{R}^{n \times p}$ is the known control matrix, $A \in \mathbb{R}^{n \times n}$ is the unknown state matrix, and $\Lambda \in \mathbb{R}^{p \times p}$ is the unknown diagonal matrix of control uncertainties with diagonal elements $\lambda_i > 0$. This uncertainty can be due to modeling errors or model control failures. It is assumed that the pair $(A, B\Lambda)$ is controllable. In addition, the unknown nonlinear vector function $f(\vec{x}) : \mathbb{R}^n \rightarrow \mathbb{R}^n$ represents the system matched uncertainty where each individual component $f_i(\vec{x})$ of $f(\vec{x})$ can be written as a linear combination of N known locally Lipschitz-continuous basis function $\phi(\vec{x})$ with unknown constant coefficients, i.e.

$$f(\vec{x}) = \vec{\theta}^T \Phi(\vec{x}) \quad (6.44)$$

where $\vec{\theta} \in \mathbb{R}^N$ is constant unknown matrix and $\Phi(\vec{x}) = [\phi_1(\vec{x}) \dots \phi_N(\vec{x})]^T$ is the known regressor vector.

A direct MIMO state feedback MRAC is required to force the system state \vec{x} to globally uniformly asymptotically track the reference model state \vec{x}_{ref} given by the **reference model**

$$\dot{\vec{x}}_{ref} = A_{ref}\vec{x}_{ref} + B_{ref}\vec{r}(t) \quad (6.45)$$

where $A_{ref} \in \mathbb{R}^n$ is chosen such that all eigenvalues are in the LHP, $B_{ref} \in \mathbb{R}^{n \times m}$ is the reference input matrix, and $\vec{r}(t) \in \mathbb{R}^p$ is the external, bounded reference command. One also requires that during tracking in an MRAC system all signals in the closed-loop system remain uniformly bounded. Note that the reference command has the same dimension as the plant control input. Thus, given any bounded command $\vec{r}(t)$, the control input $\vec{u}(t)$ needs to be chosen such that the state tracking error, $\vec{e}(t) = \vec{x}(t) - \vec{x}_{ref}(t)$, globally uniformly asymptotically tends to zero, i.e.

$$\lim_{t \rightarrow \infty} \|\vec{x}(t) - \vec{x}_{ref}\| = 0 \quad (6.46)$$

First, note that if matrices A and Λ were known, one can applied the ideal fixed-gain control law as

$$\vec{u} = K_x^T \vec{x} + K_r^T \vec{r} - \vec{\theta}^T \Phi(\vec{x}) \quad (6.47)$$

and obtain the closed-loop system

$$\dot{\vec{x}} = (A + B\Lambda K_x^T) \vec{x} + B\Lambda K_r^T \vec{r} \quad (6.48)$$

Comparing this with the desired reference dynamics, for the existence of a controller of the ideal fixed-gain form, the ideal unknown control gains, K_x and K_r , must satisfy the **matching conditions**

$$\begin{aligned} A + B\Lambda K_x^T &= A_{ref} \\ B\Lambda K_r^T &= B_{ref} \end{aligned} \quad (6.49)$$

Assuming these hold, by inspection, the ideal fixed-gain control law will result in a closed-loop system which is exactly the same as the reference model. Consequently, for any bounded reference signal input, $\vec{r}(t)$, the fixed-gain controller provides global uniform asymptotic tracking performance. It is important to note that given a general A , B , Λ , A_{ref} and B_{ref} , there is *no* guarantee that the ideal MIMO gains K_x and K_r exist such that the matching conditions are satisfied and the chosen adaptive control law may not be able to meet the design objective. However, in practice, the structure of A is known and the reference model matrices A_{ref} and B_{ref} are chosen so that the system has at least one ideal solution for K_x and K_r .

Assuming K_x and K_r do exist, consider the following control law based on the ideal fixed-gain control law, i.e.

$$\vec{u} = \hat{K}_x^T \vec{x} + \hat{K}_r^T \vec{r} - \hat{\theta}^T \Phi(\vec{x}) \quad (6.50)$$

where $\hat{K}_x \in \mathbb{R}^{n \times p}$, $\hat{K}_r \in \mathbb{R}^{p \times p}$, and $\hat{\theta} \in \mathbb{R}^{N \times n}$ are the estimates of the ideal unknown matrices K_x , K_r and $\vec{\theta}$, respectively, based on Lyapunov stability analysis. Substituting this control law into the plant dynamics results in the closed-loop system dynamics as

$$\dot{\vec{x}} = (A + B\Lambda\hat{K}_x^T) \vec{x} + B\Lambda \left(\hat{K}_r^T \vec{r} - (\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \right) \quad (6.51)$$

and subtracting the reference model from these closed-loop system dynamics, one has the closed-loop tracking error dynamics as

$$\dot{\vec{e}} = (A + B\Lambda\hat{K}_x^T) \vec{x} + B\Lambda \left(\hat{K}_r^T \vec{r} - (\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \right) - A_{ref} \vec{x}_{ref} - B_{ref} \vec{r} \quad (6.52)$$

Including the matching conditions, one has

$$\begin{aligned} \dot{\vec{e}} &= (A_{ref} + B\Lambda(\hat{K}_x - K_x)) \vec{x} - A_{ref} \vec{x}_{ref} + B\Lambda(\hat{K}_r - K_r) \vec{r} - B\Lambda(\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \\ \dot{\vec{e}} &= A_{ref} \vec{e} + B\Lambda \left((\hat{K}_x - K_x)^T \vec{x} + (\hat{K}_r - K_r)^T \vec{r} - (\hat{\theta} - \vec{\theta})^T \Phi(\vec{x}) \right) \end{aligned} \quad (6.53)$$

where defining $\Delta K_x = \hat{K}_x$, $\Delta K_r = \hat{K}_r - K_r$, and $\Delta \vec{\theta} = \hat{\theta} - \vec{\theta}$ as the parameter estimation errors, one has

$$\dot{\vec{e}} = A_{ref} \vec{e} + B\Lambda \left(\Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x}) \right) \quad (6.54)$$

Next, one can define the rates of adaptation as $\Gamma_x = \Gamma_x^T > 0$, $\Gamma_r = \Gamma_r^T > 0$, and $\Gamma_\theta = \Gamma_\theta^T > 0$. Then, consider a globally radially unbounded quadratic Lyapunov function candidate in the form

$$V(\vec{e}, \Delta K_x, \Delta K_r, \Delta \vec{\theta}) = \vec{e}^T P \vec{e} + \text{Tr} \left(\left(\Delta K_x^T \Gamma_x^{-1} \Delta K_x + \Delta K_r^T \Gamma_r^{-1} \Delta K_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \Delta \vec{\theta} \right) \Lambda \right) \quad (6.55)$$

where $P = P^T > 0$ satisfies the **algebraic Lyapunov equation**

$$P A_{ref} + A_{ref}^T P = -Q \quad (6.56)$$

for some $Q = Q^T > 0$. Then, the time derivative of V evaluated along the state trajectories can be written as

$$\dot{V} = \dot{\vec{e}}^T P \vec{e} + \vec{e}^T P \dot{\vec{e}} + 2 \text{Tr} \left(\left(\Delta K_x^T \Gamma_x^{-1} \dot{K}_x + \Delta K_r^T \Gamma_r^{-1} \dot{K}_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\theta} \right) \Lambda \right) \quad (6.57)$$

$$\begin{aligned}\dot{V} = & \left(A_{ref} \vec{e} + B\Lambda(\Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x})) \right)^T P \vec{e} \\ & + \vec{e}^T P \left(A_{ref} \vec{e} + B\Lambda(\Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x})) \right) \\ & + 2\text{Tr} \left((\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\hat{\theta}}) \Lambda \right)\end{aligned}\quad (6.58)$$

$$\begin{aligned}\dot{V} = & \vec{e}^T (A_{ref} P + P A_{ref}) \vec{e} + 2 \vec{e}^T P B \Lambda (\Delta K_x^T \vec{x} + \Delta K_r^T \vec{r} - \Delta \vec{\theta}^T \Phi(\vec{x})) \\ & + 2\text{Tr} \left((\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x + \Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r + \Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\hat{\theta}}) \Lambda \right)\end{aligned}\quad (6.59)$$

$$\begin{aligned}\dot{V} = & -\vec{e}^T Q \vec{e} + \left(2 \vec{e}^T P B \Lambda \Delta K_x^T \vec{x} + 2\text{Tr} \left(\Delta K_x^T \Gamma_x^{-1} \dot{\hat{K}}_x \Lambda \right) \right) \\ & + \left(2 \vec{e}^T P B \Lambda \Delta K_r^T \vec{r} + 2\text{Tr} \left(\Delta K_r^T \Gamma_r^{-1} \dot{\hat{K}}_r \Lambda \right) \right) \\ & + \left(-2 \vec{e}^T P B \Lambda \Delta \vec{\theta}^T \Phi(\vec{x}) + 2\text{Tr} \left(\Delta \vec{\theta}^T \Gamma_\theta^{-1} \dot{\hat{\theta}} \Lambda \right) \right)\end{aligned}\quad (6.60)$$

Next, one can use the vector trace identity $\vec{a}^T \vec{b} = \text{Tr}(\vec{b} \vec{a}^T)$ to form

$$\begin{aligned}\dot{V} = & -\vec{e}^T Q \vec{e} + 2\text{Tr} \left(\Delta K_x^T \left(\Gamma_x^{-1} \dot{\hat{K}}_x + \vec{x} \vec{e}^T P B \right) \Lambda \right) \\ & + 2\text{Tr} \left(\Delta K_r^T \left(\Gamma_r^{-1} \dot{\hat{K}}_r + \vec{r} \vec{e}^T P B \right) \Lambda \right) + 2\text{Tr} \left(\Delta \vec{\theta}^T \left(\Gamma_\theta^{-1} \dot{\hat{\theta}} - \Phi(\vec{x}) \vec{e}^T P B \right) \Lambda \right)\end{aligned}\quad (6.61)$$

Thus, if the **Direct MIMO adaptive laws** are chosen as

$$\begin{aligned}\dot{\hat{K}}_x &= -\Gamma_x \vec{x} \vec{e}^T P B \\ \dot{\hat{K}}_r &= -\Gamma_r \vec{r}(t) \vec{e}^T P B \\ \dot{\hat{\theta}} &= \Gamma_\theta \Phi(\vec{x}) \vec{e}^T P B\end{aligned}\quad (6.62)$$

Then,

$$\dot{V} = -\vec{e}^T Q \vec{e} \leq 0 \quad (6.63)$$

Therefore, the closed-loop error dynamics are uniformly stable. So, the tracking error $\vec{e}(t)$ and the parameter estimation errors $\Delta K_x(t)$, $\Delta K_r(t)$, and $\Delta \theta(t)$ are uniformly bounded and so are the parameter estimates $\hat{K}_x(t)$, $\hat{K}_r(t)$, and $\hat{\theta}(t)$. Since $\vec{r}(t)$ is bounded and A_{ref} has all LHP eigenvalues, then \vec{x}_{ref} and \vec{x}_{ref}' are bounded. Hence, the system state $\vec{x}(t)$ is uniformly bounded, the control input $\vec{u}(t)$ is bounded, and $\vec{x}(t)$ is bounded, and thus, $\vec{e}(t)$ is bounded. Furthermore, the second time derivative of $V(t)$

$$\ddot{V} = -2 \vec{e}^T Q \vec{e} \quad (6.64)$$

is bounded, and so $\dot{V}(t)$ is uniformly continuous. Since, in addition, $V(t)$ is lower bounded and $\dot{V}(t) \leq 0$, then using Barbalat's lemma $\lim_{t \rightarrow \infty} \dot{V}(t) = 0$. Thus, it has been formally proven that the state tracking error $\vec{e}(t)$ tends to the origin globally, uniformly, and asymptotically, i.e.

$$\lim_{t \rightarrow \infty} \|\vec{x}(t) - \vec{x}_{ref}(t)\| = 0 \quad (6.65)$$

Lastly, note that the tuning knobs in the MIMO case are Γ_x , Γ_r , and Γ_θ as before, but also Q for the algebraic Lyapunov equation, all of which must be symmetric, positive definite matrices.

Derivative-Free Direct MRAC

Indirect MRAC

6.4 L_1 Adaptive Control

6.5 Concurrent Learning Adaptive Control

Integrated Concurrent Learning Adaptive Control

Part II

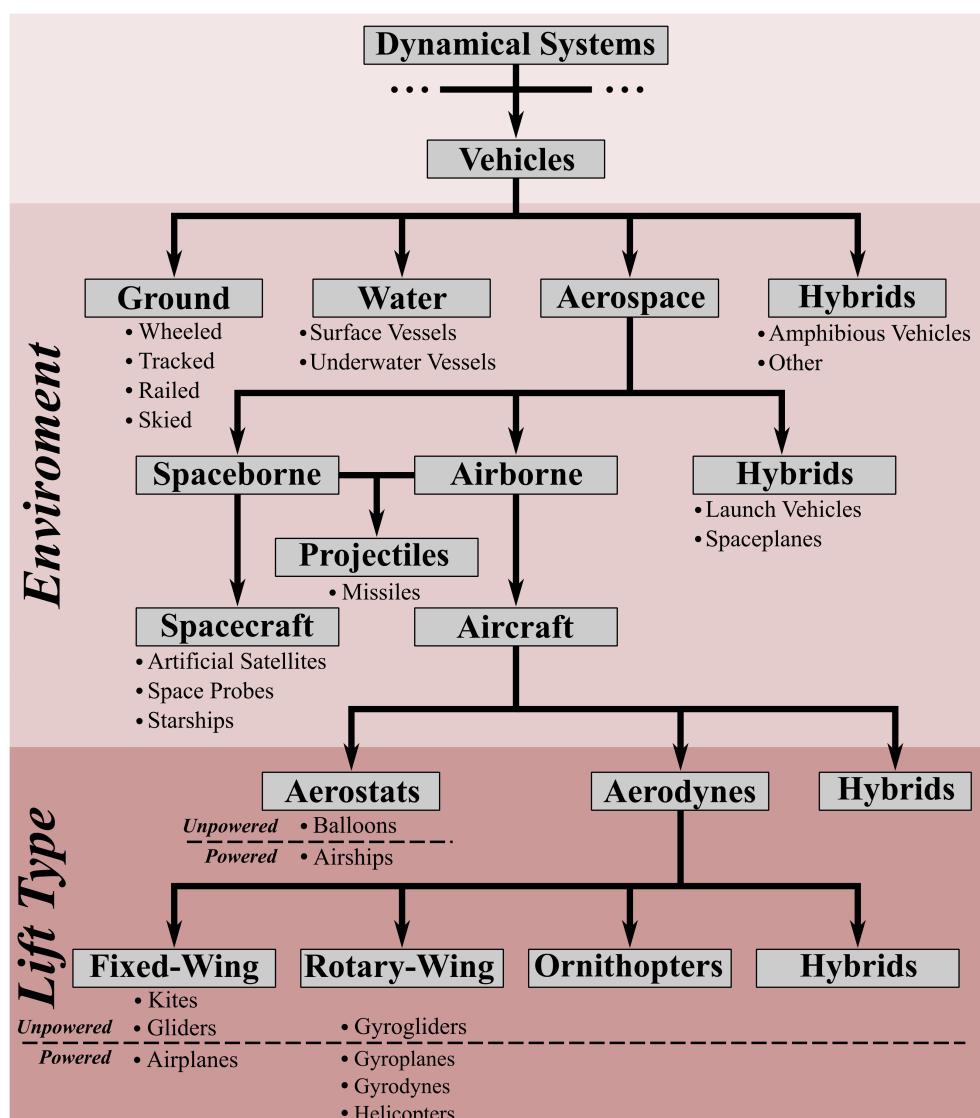
Aerospace Vehicle Dynamics and Control Systems

CHAPTER

7

Aerospace Vehicle Attitude and Steady-Flight Models

7.1 Introduction to Aerospace Vehicles



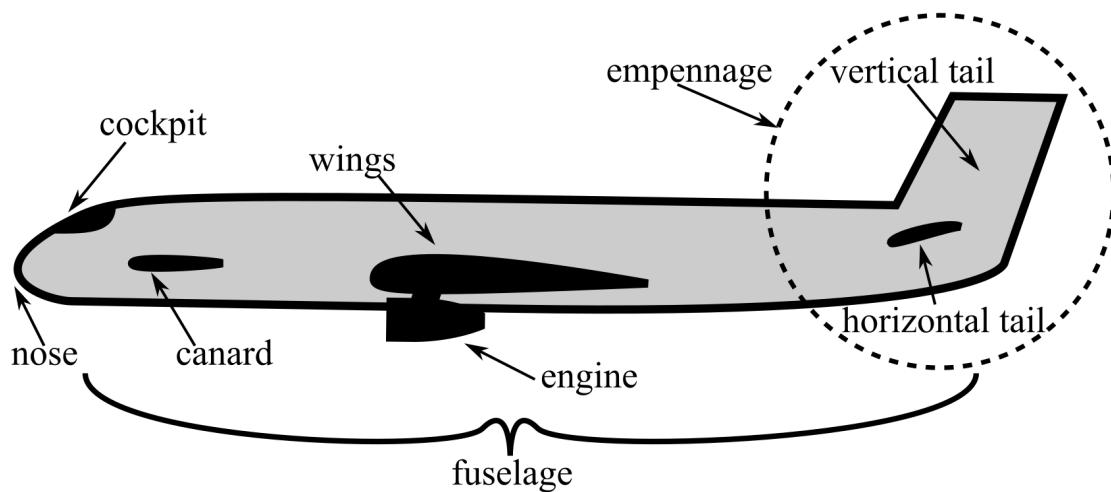
A **vehicle** is defined as a dynamical system that transports a payload, e.g. cargo, people, munitions, sensors. Vehicles are typically classified by the environment in which they operate, i.e. ground, water, flying, or amphibious. **Aerospace vehicles** are a category of vehicle encompassing aircraft, spacecraft, airborne and spaceborne projectiles, and air/space hybrids. These are also known as **flight vehicles**. There are many different types of aerospace vehicles. These are separated into aerostats (e.g. balloons, airships) and aerodynes which includes fixed-wing aircraft (e.g. gliders, airplanes), rotary-wing aircraft (e.g. gyrogliders, helicopters, autodynes, gyrodyne), spacecraft (e.g. artificial satellites, space probes, starships), and hybrid aerospace vehicle that operate in air and space (e.g. launch vehicles, spaceplanes). However, it is important to note that the dynamics and control concepts in this part of the textbook can be applied to *any* dynamical system to varying degrees as one diverges further away from aerospace vehicles.

Vehicle dynamics and control is the study of the changes in a vehicle's motion due to the forces and moments applied to and by that vehicle in its environment. The motion of a vehicle is typically described by its position, orientation/attitude, and velocity. When applied in particular to aerospace vehicle, this study is called **flight dynamics and control (FDC)** and is the subject of this part of the textbook. The important differentiating factor in FDC from other vehicles is that aerospace vehicle are affected by forces and moments from only *four* external sources: gravity, propulsion, aerodynamics, and solar radiation, whereas ground and water vehicles experience ground forces and hydrodynamics, respectively. Modeling the forces and moments from these four external sources are the differentiating factors between aerospace vehicle dynamics and other physical object dynamics.

Aerospace vehicles are designed so that the **operator** of the aerospace vehicle, also known as the **pilot**, can affect the aerodynamic and propulsive forces and moments imparted on the aerospace vehicle in order to control or affect the vehicle's motion, i.e. its dynamics. The control of a aerospace vehicle can be performed completely by a human, also known as **manual control**, completely by a computer, also known as **automatic control**, which constitutes an **autopilot**, or partially from both, also known as **semi-automatic control**. In this part of the textbook, classical control theory will be introduced and applied to the control of aerospace vehicle.

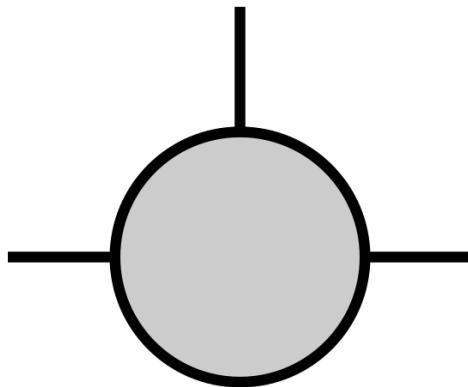
Fixed-Wing Aircraft Anatomy

The basic components of a conventional **airplane**, i.e. a powered fixed-wing aircraft, are shown in the following diagram



The **nose** is the front of the airplane and houses the **cockpit** where the pilot(s) and/or other operators are located. The **fuselage** is the tubular structure of the airplane that houses the payload, fuel/batteries, and **avionics**, i.e. the aviation electronics. The **wings** extend horizontally out of the fuselage and generate the majority of the lift force to overcome the vehicle's weight and fly. Some airplanes have two wings, either as **tandem wings** which are located as front-and-back wings or as stacked wings called a **biplane**. The **engines** produce the thrust forces, with a possible **propeller** creating a swirling slipstream using rotating airfoil-section blades, which are controlled by the pilot. These can be mounted on the wings, empennage, or nose.

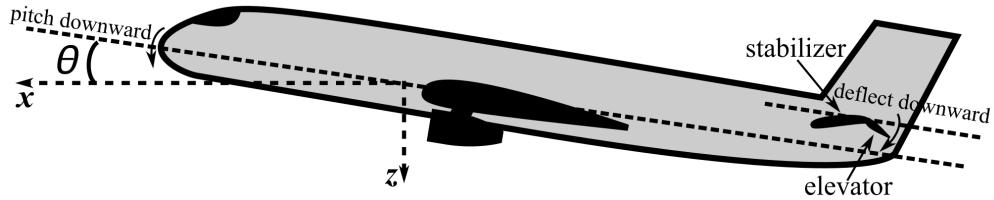
The **empennage** is the rear section of an airplane and creates aerodynamic effects to stabilize and steer the airplane in the direction of flight. This airplane shows the **conventional tail**.



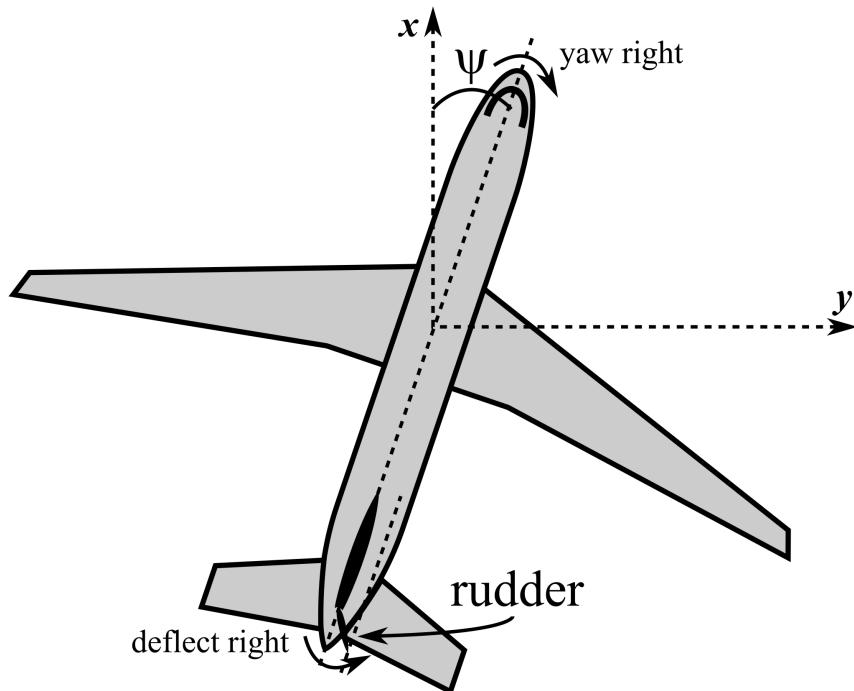
The **vertical tail** is composed of a static **vertical stabilizer** and a dynamic **rudder** which produce a horizontal left or right side force to maintain vertical stability and steer the airplane. The **horizontal tail** is composed of a static **horizontal stabilizer** and a dynamic **elevator** which produce a vertical up or down lift force to maintain horizontal stability and steer the airplane. Some airplanes may be **tailless** which do

not have any other horizontal lifting surface besides its main wing, i.e. no horizontal tail, canard or tandem wing. They may or may not have a vertical tail. These airplanes use compound ailerons/elevator control surfaces called **elevons** or **tailerons**. Furthermore, a **flying wing** is a tailless airplane that also does not have a distinct fuselage. Some airplanes allow the entire horizontal tail to move which forms a compound horizontal stabilizer/elevator called a **stabilator**. Some airplanes may use a **canard**, i.e. a forward horizontal stabilizer and elevator, either in addition or instead of a rear horizontal tail.

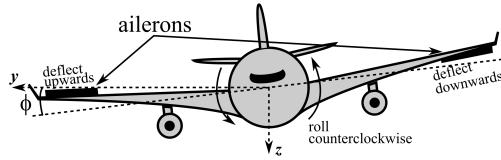
Fixed-wing aircraft generally have three major control surfaces that allow the pilot or autopilot to affect the aerodynamic forces on the aircraft thereby altering its motion in a controlled manner. The **elevator** is mounted on trailing edge of horizontal tail and is used as the primary pitch angle, θ , control input as shown in the following graphic where a deflection angle, δ_e , upwards pushes the airplane's nose up.



The **rudder** is mounted on trailing edge of the vertical tail and is used as the primary yaw angle, ψ , control input as shown in the following graphic where a deflection angle, δ_r , right pushes airplane's nose right.

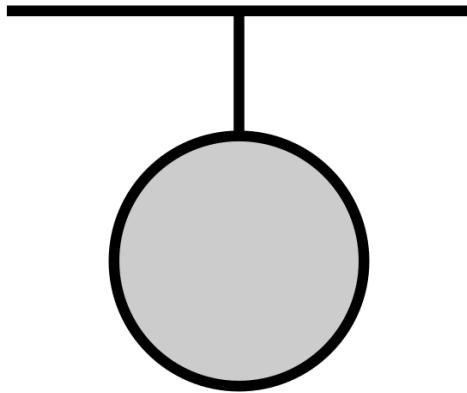


The **ailerons** are a differential pair of surfaces, i.e. if one deflects up, the other deflects down, mounted on trailing edge of wings near the tips and are used as the primary roll angle, ϕ , control input as shown in the following graphic where the airplane rolls to the side that deflects up at some deflection angle, δ_a .

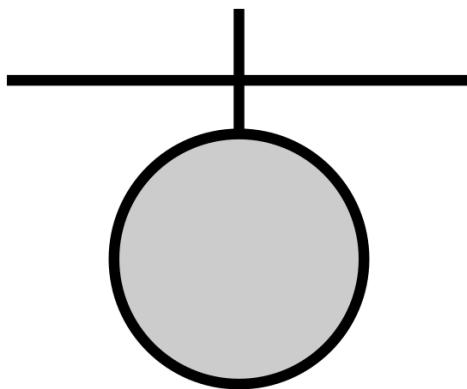


Depending on the tail, some airplanes may have multiple of these control surfaces, e.g. H-tails have two rudders.

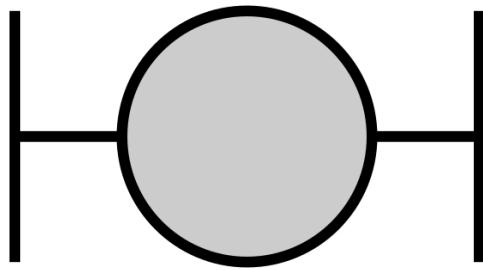
There are also a wide variety of empennage configurations, e.g. a **T-tail**



a **cruciform tail**

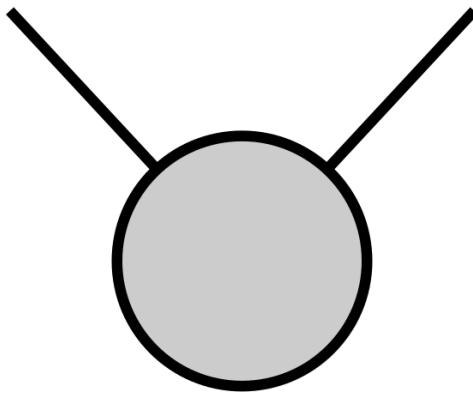


or an **H-tail**, also known as a **twin tail**.

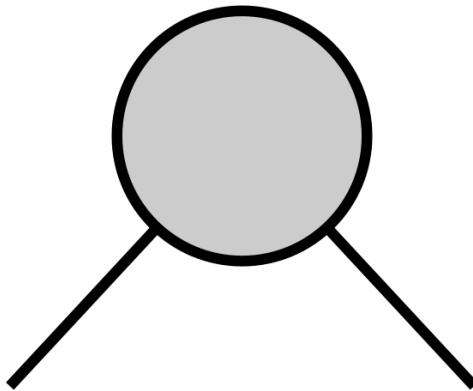


These twin tails may be mounted on the fuselage, two booms extending from the wing, or the wings themselves.

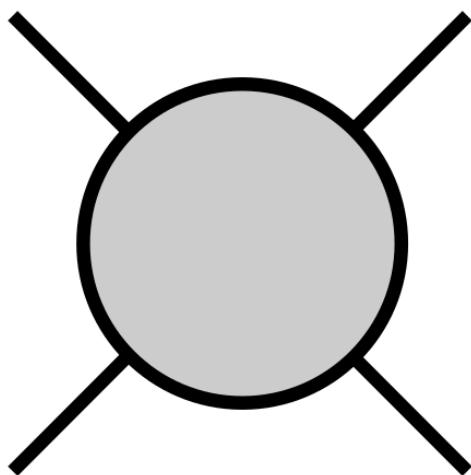
In addition, one may also use a compound horizontal/vertical tail empennage, e.g. a **V-tail**



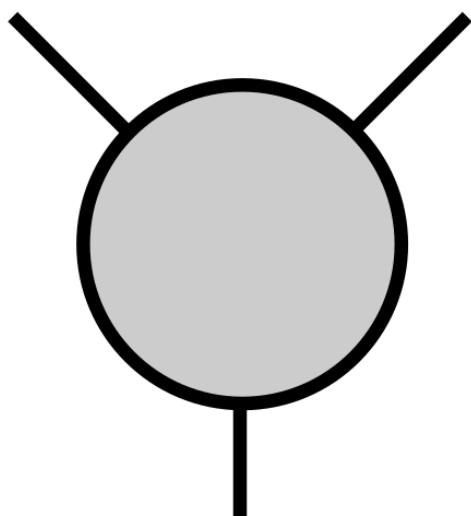
inverted V-tail



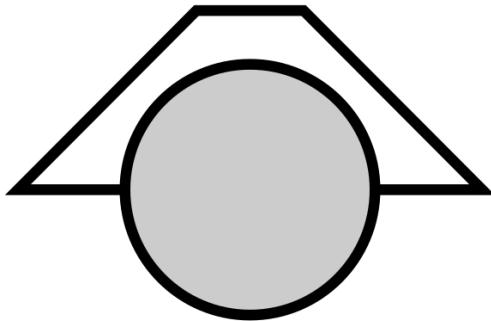
X-tail



Y-tail



or an **A-tail**

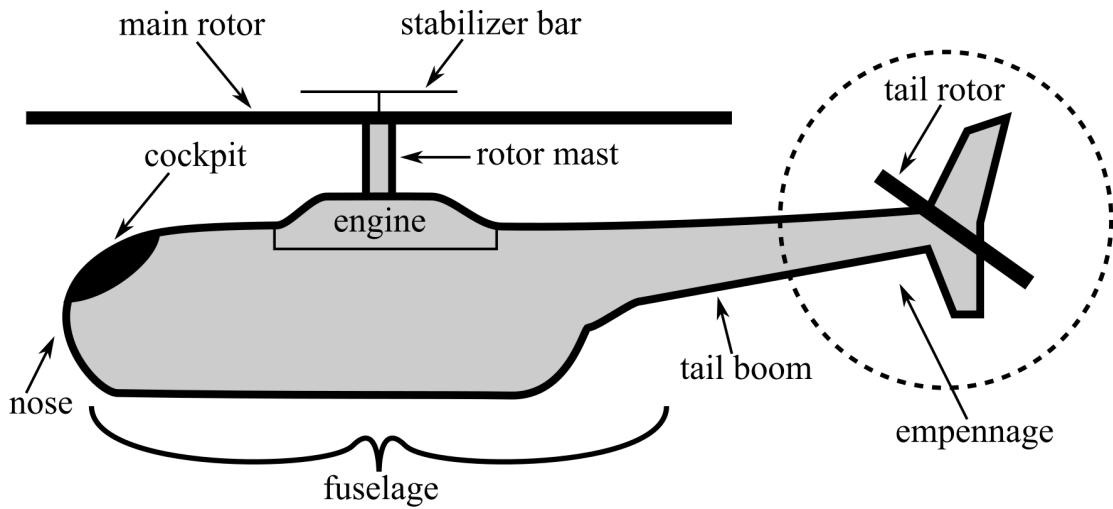


which are typically mounted on twin booms. These empennage configurations notably use compound rudder and elevator control surfaces called a **ruddervator**.

Many airplanes also have secondary control surfaces that can be controlled to change the airplane aerodynamics and thus alter its motion. These are typically grouped into two different categories. **Spoilers** are hinged flat plates attached to wing facing oncoming airflow. These “spoil” the lift on the wing when rotated up from the surface of the wing and thus are typically used as speed brakes, but sometimes can be used for roll control with or without ailerons. **Slats** and **flaps** are mechanized leading and trailing edges of wings, respectively, that can be extended/retracted from the nominal wing shape, thereby changing the airfoil cross-sectional shape and effectively increasing/decreasing the lift potential of the wing with higher/lower induced drag. As such, these are primarily extended during takeoff and landing and are not dynamically active control surfaces, i.e. they are either extended or retracted for different flight conditions and tasks.

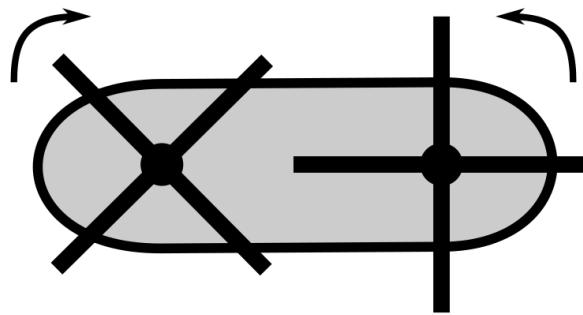
Rotary-Wing Aircraft Anatomy

The basic components of a conventional single-rotor **helicopter**, i.e. powered rotary-wing aircraft without propellers, are shown in the following diagram

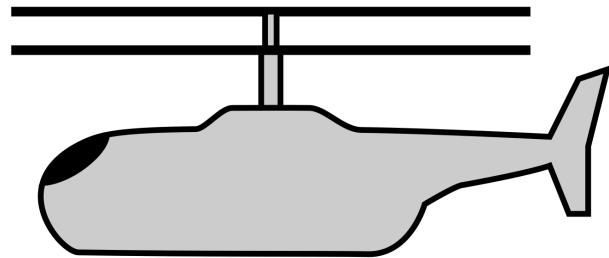


The **nose** is the front of the helicopter and houses the **cockpit** where the pilot(s) and/or other operators are located. The **fuselage** is the structure of the helicopter beneath the main rotor that houses the payload, fuel/batteries, and avionics. The **main rotor** provides lift as the blades rotate to overcome the vehicle's weight and fly. The lift can be changed by the rotor's angular velocity or the rotor's angle of attack, either **collectively** or **cyclically**, i.e. at different parts of its rotation cycle. This angle of attack is acutated through a **swashplate** mounted on the **rotor mast** or through rotor servo flaps. This adjustment to the relative blade angle of attack allows for forward and lateral movement of the helicopter. The optional **stabilizer bar** sits above and across the main rotor blade and dampens unwanted vibrations in the main rotor, helping to stabilize the helicopter in all flight conditions. The second **tail rotor** allows for directional control of the helicopter and is required in order to negate the angular momentum produced by the main rotor blades on the helicopter.

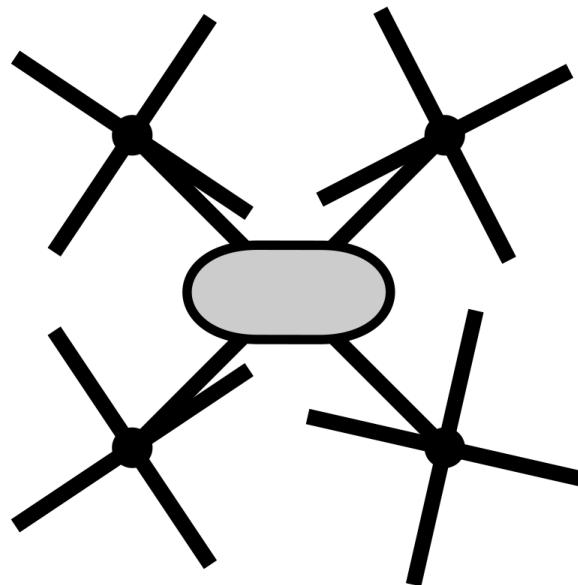
Thus, helicopters are designed to cancel out the rotary-wing angular momentum through tail rotors as shown or with **counter-rotating rotors**, i.e. pairs of rotors which rotate in opposite directions in order to counter each other's angular momentum, e.g. tandem rotor configuration



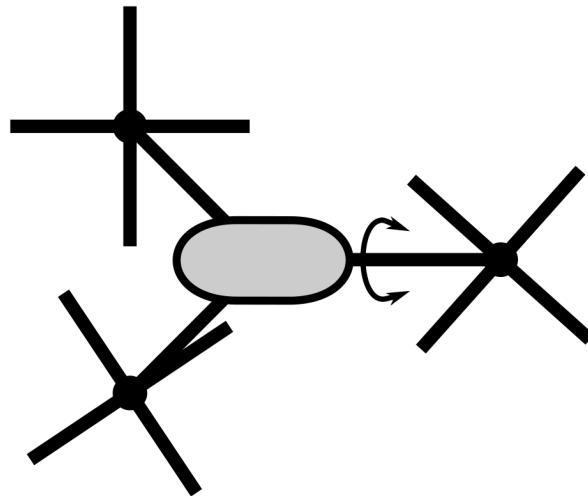
coaxial rotors configuration



or multirotor (≥ 3) configurations, e.g. a quadrotor or 4-rotor configuration

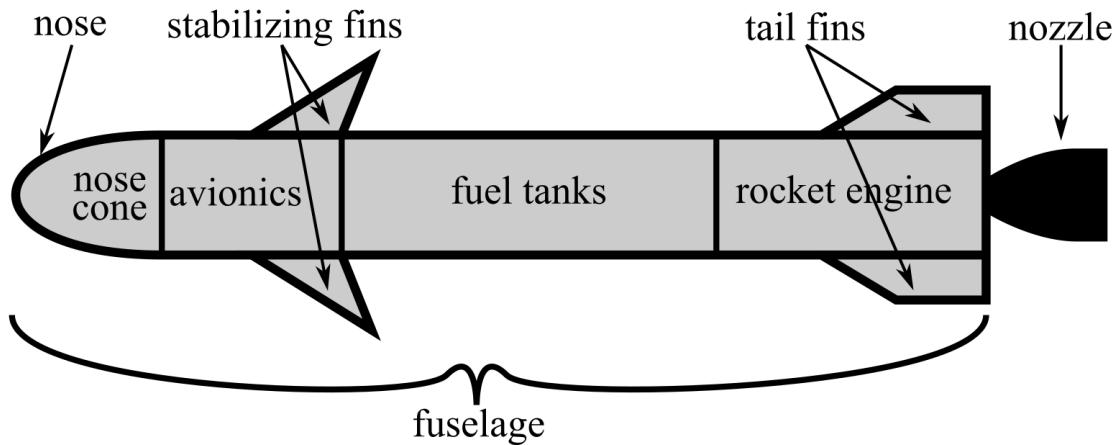


Notably, with ≥ 4 rotors, one does not require any rotor angle of attack control, as the angular velocities of each rotor allow for total motion control. For a **tricopter**, a 3-rotor configuration, one of the rotors is typically allowed to rotate about its arm to provide a thrust force with some horizontal component.



Launch Vehicle Anatomy

The basic components of a conventional launch vehicle design are shown in the following diagram

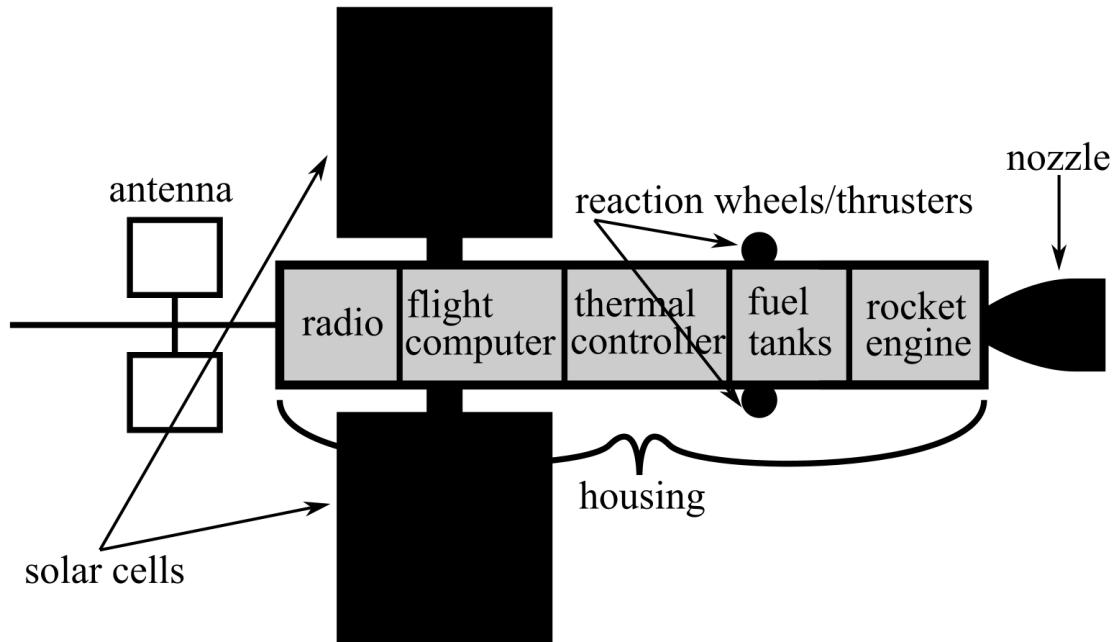


The **nose cone** is the front of the launch vehicle and houses the **fairing** typically for the payload which may include crew. The **fuselage** is the tubular structure of the launch vehicle that houses the avionics, fuel, oxidizer, and rocket engine. The **fins**, both stabilizing and tail, produce an up or down lift force to steer the launch vehicle. Some launch vehicles may have only tail fins. These fins also may have control surfaces on them called **rudders**. The **rocket engine** projects its reaction mass through the **nozzle** to produce the thrust force to overcome the vehicle's weight and fly. Some launch vehicle nozzle's may have thrust vectoring capabilities for control as well.

Artificial Satellite Anatomy

Satellites are objects that travel in **orbits** around celestial bodies. Satellites can be natural, e.g. moons, planets, or **artificial satellites**, i.e. human-made spacecraft designed for orbital flight.

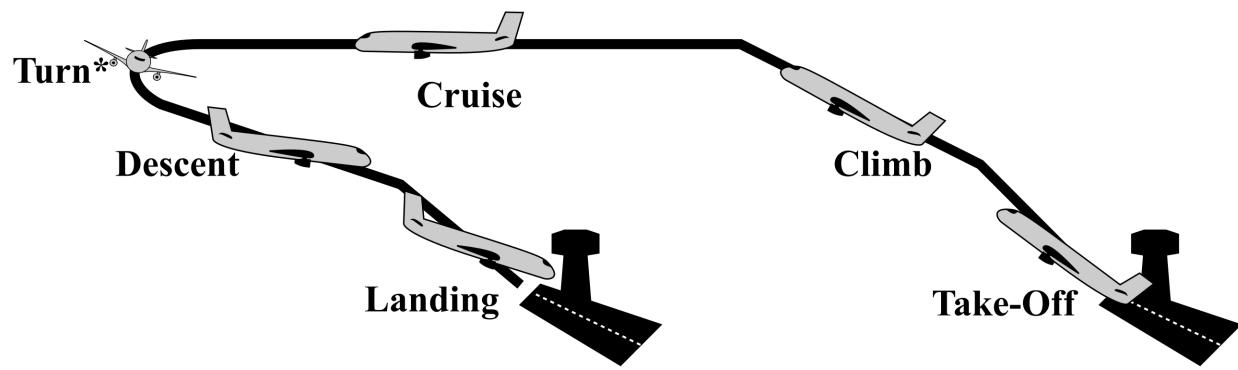
The basic components of a conventional artificial satellite design are shown in the following diagram



The **housing** houses the rocket engine, fuel tanks, thermal controller, flight computer, the **radio** which communicates between the satellite and the ground control station and/or other satellites, and the **antenna** which converts the electrical signal to radio waves and vice versa. **Reaction wheels or thrusters** are placed strategically on the satellite housing to provide control inputs for maintaining orbit stabilization. Finally the satellite is powered using batteries and **solar cells** mounted on the satellite.

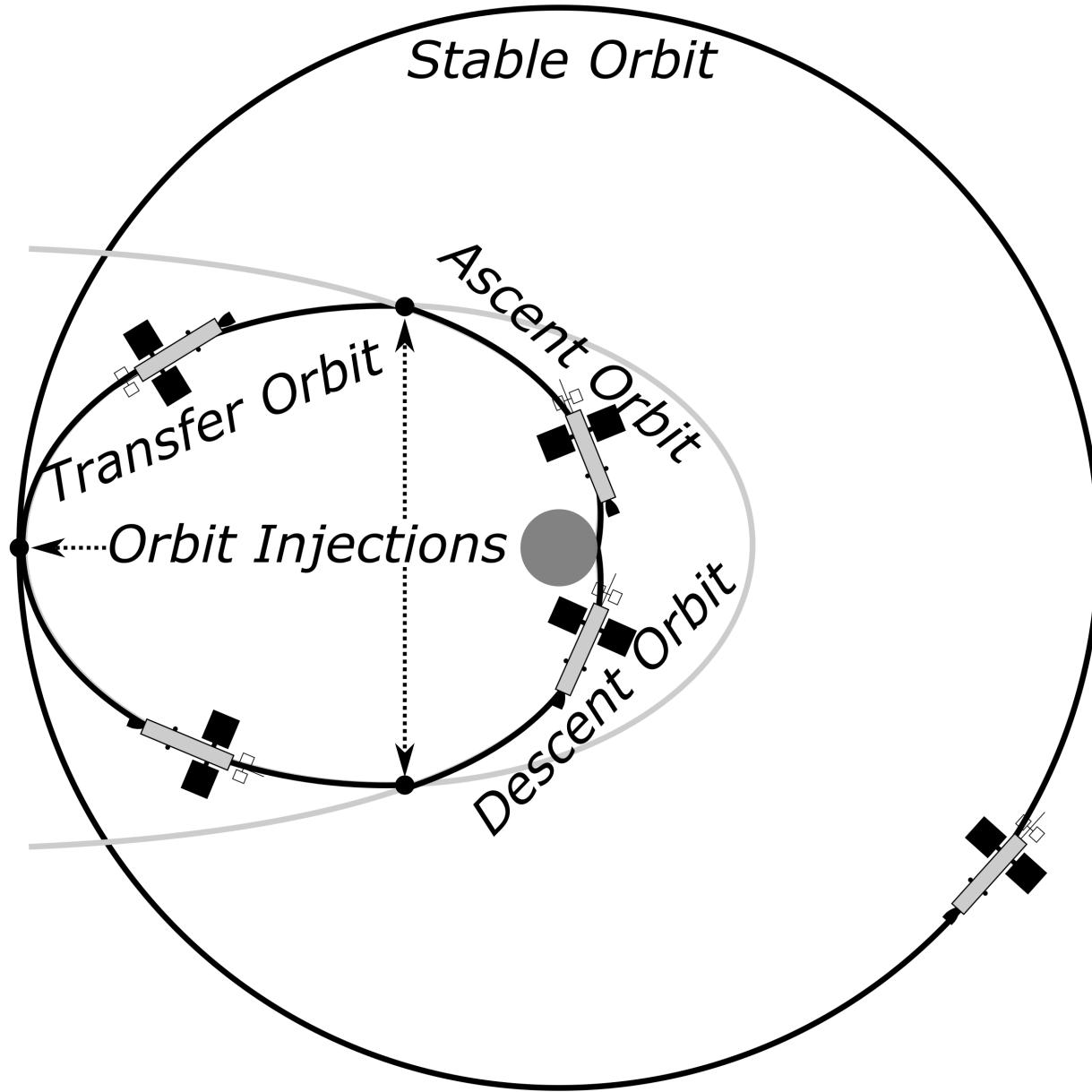
Phases of Flight

The five phases of aircraft flight can generally be described as takeoff, climb, cruise, descent, landing as shown in the following graphic.



where each phase could include maneuvers such as coordinated turns. The aircraft will experience different flight conditions and may be configured differently at each phase, e.g. for takeoff and landing of airplanes, flaps may be deployed and the landing gear will be extended down, which will directly impact the vehicle dynamics.

The four phases of spacecraft flight can generally be described as launch, orbit(s), reentry, and optionally landing. The orbits may consist of ascent orbit, transfer orbit, stable orbit, and a descent orbit, as shown in the following graphic for an Earth-centered mission.



where each orbital change is performed using an **orbit injection**, also known as an **orbit insertion**, which is a type of orbital maneuver. An **orbital maneuver**, also known as a **burn**, is the use of propulsion system to change the orbit of a spacecraft. When not undergoing an orbital maneuver, a spacecraft is in **coast**.

There are many other orbital maneuvers besides orbit insertions, often distinguished by either as impulsive, non-impulsive, or gravity assist, each producing some amount of change in velocity called the Δv . An **impulsive maneuver** uses the burn to generate a particular Δv almost instantaneously, e.g. **Oberth**

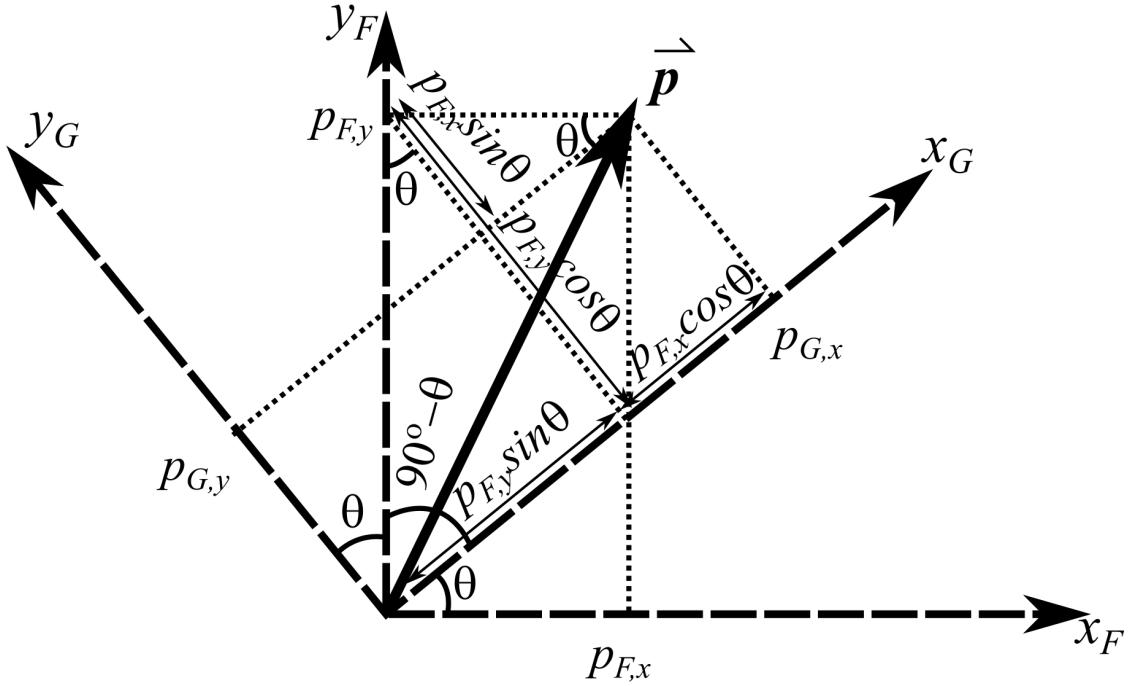
maneuver use burn close to gravitational bodies to increase final velocity. A **non-impulsive maneuver**, a.k.a. **finite burn** uses the burn to change the momentum slowly over long time by expending a low amount of propulsion. A **gravity assist maneuver** does not use a burn, but instead uses gravitational bodies to accelerate vehicle instead. Other types of orbital maneuvers include changing the orbital parameters, e.g. an **orbit inclination maneuver** changes a spacecraft's inclination angle and an **orbit phasing maneuver** adjusts a spacecraft's true anomaly. Lastly, a **rendezvous maneuver** is the coordination of two spacecraft, called the “chaser” and the “target,” arriving at the same orbit and approaching to a very close relative distance. Often, these rendezvous maneuvers are accompanied by other maneuvers such as coordinated operations in close proximity, docking, and undocking. Thus, these are often grouped together as **rendezvous, proximity operations, docking, and undocking (RPODU)** procedures for spacecraft.

7.2 Reference Frames and Rotations

In order to derive the dynamical system models for flight vehicles, one uses various coordinate systems centered at different origins, known as **reference frames**. A reference frame F is uniquely described by its origin point O_F and three axes $x_F - y_F - z_F$. Furthermore, when discussing dynamics, one must be able to transform the vehicle state between different reference frames. This section will discuss vector transformations for two- and three-dimensional vectors referenced to the same origin point, also known as **reference frame rotations**. Notably, this is a related, but different concept than the active rotation of vectors.

Two-Dimensional Reference Frame Rotations

Consider the vector \vec{p} in frames F and G , expressed as $\vec{p}_F = [p_{F,x} \ p_{F,y}]^T$ and $\vec{p}_G = [p_{G,x} \ p_{G,y}]^T$ with the same origin.



By trigonometry,

$$\vec{p}_G = \begin{bmatrix} p_{G,x} \\ p_{G,y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} p_{F,x} \\ p_{F,y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \vec{p}_F \quad (7.1)$$

or

$$\vec{p}_G = \begin{bmatrix} p_{G,x} \\ p_{G,y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} p_{F,x} \\ p_{F,y} \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \vec{p}_F \quad (7.2)$$

which can be defined as

$$\vec{p}_G = C_{G \leftarrow F}(\theta) \vec{p}_F \quad (7.3)$$

where $C_{G \leftarrow F}(\theta)$ is the **two-dimensional direction cosine matrix (DCM)** from frame F to G by the angle θ . This name originates from the identity, $\sin \theta = \cos(90^\circ - \theta)$ and $-\sin \theta = \cos(90^\circ + \theta)$, where one can rewrite

$$C_{G \leftarrow F} = \begin{bmatrix} \cos \theta & \cos(90^\circ - \theta) \\ \cos(\theta - 90^\circ) & \cos \theta \end{bmatrix} = \begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} \quad (7.4)$$

where each element, $C_{i,j}$, corresponds to the cosine of the angle between the i -axis of the G frame and the j -axis of the F frame. This can also be seen by inspection of the previous figure.

To transform back, one can form the inverse DCM by using the negative angle, i.e.

$$\vec{p}_F = \begin{bmatrix} \cos(-\theta) & \sin(-\theta) \\ -\sin(-\theta) & \cos(-\theta) \end{bmatrix} \vec{p}_G = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \vec{p}_G \quad (7.5)$$

or

$$\vec{p}_F = C_{F \leftarrow G}(-\theta) \vec{p}_G \quad (7.6)$$

which by inspection can also be seen to be

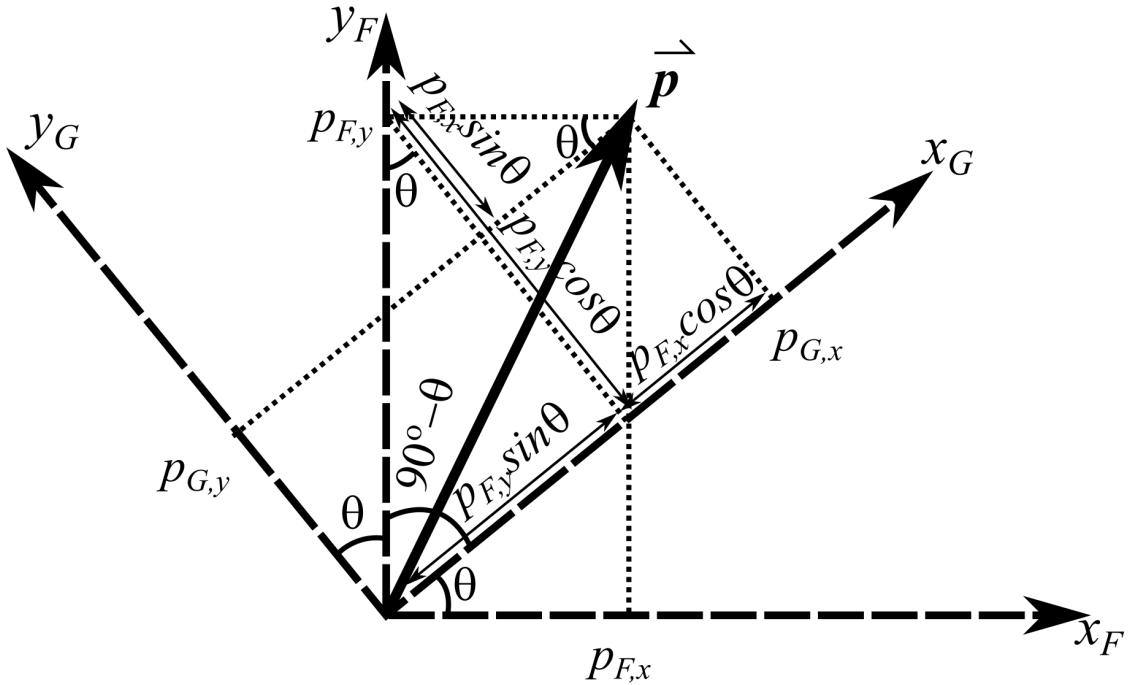
$$\vec{p}_F = C_{G \leftarrow F}^T(\theta) \vec{p}_G \quad (7.7)$$

which implies the property

$$C_{G \leftarrow F}^{-1}(\theta) = C_{G \leftarrow F}^T(\theta) \quad (7.8)$$

for direction cosine matrices.

The DCM is also known as a **passive rotation matrix**, as opposed to an **active rotation matrix** of the vector \vec{p} by an angle θ to obtain a new vector \vec{p}' .



In this case, an active rotation in the reference frame F would be given by

$$\vec{p}'_F = R \vec{p}_F = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix} \vec{p}_F = C_{G \leftarrow F}^T \vec{p}_F \quad (7.9)$$

so it is important to note the context of “rotation matrix” when reading sources on rotational kinematics.

Next, consider if the vector and the rotation angle are changing in time, then one can represent the time derivative of the vector coordinates, $\vec{v}(t) = \frac{d\vec{p}}{dt}(t)$, also known as the “inertial” **two-dimensional linear velocity** in the static “inertial” frame F or the rotating frame G . In the latter case, one must first calculate the following

$$\vec{v}_F(t) = C_{F \leftarrow G} \vec{v}_G(t) = \frac{d}{dt} (C_{F \leftarrow G}(t) \vec{p}_G(t)) = \frac{d}{dt} \left(\begin{bmatrix} \cos \theta(t) & -\sin \theta(t) \\ \sin \theta(t) & \cos \theta(t) \end{bmatrix} \vec{p}_G(t) \right) \quad (7.10)$$

which by the product rule provides

$$C_{F \leftarrow G} \vec{v}_G(t) = C_{G \leftarrow F}^T \dot{\vec{p}}_G(t) + \begin{bmatrix} -\sin \theta(t) & \cos \theta(t) \\ -\cos \theta(t) & -\sin \theta(t) \end{bmatrix} \dot{\theta}(t) \vec{p}_G(t) \quad (7.11)$$

where $\dot{\vec{p}}$ is simply the partial time derivative of the vector itself, i.e.

$$\dot{\vec{p}}_G(t) = \frac{\partial}{\partial t} \vec{p}_G(t) \quad (7.12)$$

and not the rotating frame.

By rearranging the second term and multiplying by $C_{G \leftarrow F}$, one can write

$$C_{F \leftarrow G} \vec{v}_G(t) = \begin{bmatrix} \cos \theta(t) & -\sin \theta(t) \\ \sin \theta(t) & \cos \theta(t) \end{bmatrix} \dot{\vec{p}}_G(t) + \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \cos \theta(t) & -\sin \theta(t) \\ \sin \theta(t) & \cos \theta(t) \end{bmatrix} \dot{\theta}(t) \vec{p}_G(t) \quad (7.13)$$

Finally, by defining $\omega(t) = \dot{\theta}(t)$ as the **two-dimensional angular velocity** and multiplying by $C_{G \leftarrow F}$ on both sides, one has

$$\vec{v}_G(t) = \dot{\vec{p}}_G(t) + \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \vec{p}_G(t) \quad (7.14)$$

where \vec{v}_G is the **two-dimensional linear velocity** in the frame G and the **two-dimensional angular velocity tensor** is

$$[\omega(t)]_\times = \begin{bmatrix} 0 & -\omega(t) \\ \omega(t) & 0 \end{bmatrix} \quad (7.15)$$

and, notably, one has the relationship

$$\dot{C}_{F \leftarrow G} = [\omega]_\times C_{F \leftarrow G} \quad (7.16)$$

and

$$\dot{C}_{F \leftarrow G} = -C_{F \leftarrow G} [\omega]_\times \quad (7.17)$$

This derivative rule applies for any vector quantity defined in the G frame. Similarly, one can define the derivative of the linear velocity as

$$\frac{d}{dt} \vec{v}_G(t) = \vec{a}_G(t) = \dot{\vec{v}}_G(t) + \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \vec{v}_G(t) \quad (7.18)$$

where $\vec{a}_G(t)$ is the **two-dimensional acceleration**, i.e. $\frac{d^2}{dt^2} \vec{p}_G(t)$. To compute the second time derivative of the vector coordinates in the rotating frame G for the position vector, as represented in the rotating frame, then another instance of the product rule for both terms can be employed to obtain

$$\frac{d^2 \vec{p}}{dt^2}{}_G(t) = \frac{d \vec{v}}{dt}{}_G(t) = \vec{a}_G(t) \quad (7.19)$$

$$\vec{a}_G(t) = \frac{d}{dt} (\dot{\vec{v}}_G(t)) + \frac{d}{dt} \left(\begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \vec{v}_G(t) \right) \quad (7.20)$$

$$\vec{a}_G(t) = \ddot{\vec{p}}_G(t) + \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \dot{\vec{p}}_G(t) + \begin{bmatrix} 0 & \dot{\omega}(t) \\ -\dot{\omega}(t) & 0 \end{bmatrix} \vec{p}_G(t) + \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \frac{d \vec{p}}{dt}{}_G(t) \quad (7.21)$$

and using the previous formula for $\frac{d}{dt} \vec{p}_G(t)$ and the substitution $\alpha = \dot{\omega}$ as the **two-dimensional angular acceleration**, one has

$$\begin{aligned}\vec{a}_G(t) &= \ddot{\vec{p}}_G(t) + 2 \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \dot{\vec{p}}_G(t) + \begin{bmatrix} 0 & \alpha(t) \\ -\alpha(t) & 0 \end{bmatrix} \vec{p}_G(t) \\ &\quad + \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \vec{p}_G(t)\end{aligned}\tag{7.22}$$

where

$$2 \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \dot{\vec{p}}_G(t)\tag{7.23}$$

is the **two-dimensional Coriolis acceleration**,

$$\begin{bmatrix} 0 & \alpha(t) \\ -\alpha(t) & 0 \end{bmatrix} \vec{p}_G(t)\tag{7.24}$$

is the **two-dimensional Euler acceleration**, and

$$\begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \begin{bmatrix} 0 & \omega(t) \\ -\omega(t) & 0 \end{bmatrix} \vec{p}_G(t)\tag{7.25}$$

is the **two-dimensional centrifugal acceleration**. Collectively, these are known as **fictitious accelerations** and produce **fictitious forces**.

Three-Dimensional Reference Frame Rotations

For three-dimensional rotations, one has four different methods for specifying the rotation between two different reference frames. The first representation is the **axis-angle** representation defined by **Euler's rotation theorem** states that any rotating frame possesses an instantaneous axis of rotation. Thus, the first representation specifies the instantaneous axis of rotation, also known as the **Euler axis**, as $\vec{e} = [\cos(e_x) \cos(e_y) \cos(e_z)]^T$, of unit length as well as the **rotation angle**, θ , about that axis. Note that in two-dimensions, the Euler axis was the normal to the two-dimensional plane, i.e. a hypothetical “z-axis” and the angle was θ . However, in three-dimensions, the Euler axis is arbitrary. As one can describe the passive rotation vector between the frames as $\vec{\theta} = \theta \vec{e}$, one can define a three-dimensional rotating reference frames **three-dimensional angular velocity** as $\vec{\omega}(t) = \dot{\vec{\theta}}(t) = \omega \vec{e}$, and the **three-dimensional angular acceleration**, $\vec{\alpha}(t) = \dot{\vec{\omega}}(t) = \ddot{\vec{\theta}}(t) = \alpha \vec{e}$. Here the magnitude of the three-dimensional angular velocity and acceleration is consistent with the two-dimensional case.

A mathematically convenient and vectorized representation of the Euler axis and rotation angle is the second representation known as the **Euler symmetric parameters**, also known as a **passive rotation quaternion**, which can be defined in several different ways resulting in different rigid-body kinematics, but this representation is beyond the scope of this textbook.

Similar to the two-dimensional rotation matrix, the third representation is to use a three-dimensional rotation matrix also known as the **three-dimensional direction cosine matrix (DCM)**, $C_{G \leftarrow F}$, i.e.

$$\vec{v}_G = C_{G \leftarrow F} \vec{v}_F\tag{7.26}$$

which is often denoted component-wise as

$$C_{G \leftarrow F} = \begin{bmatrix} C_{11} & C_{12} & C_{13} \\ C_{21} & C_{22} & C_{23} \\ C_{31} & C_{32} & C_{33} \end{bmatrix} \quad (7.27)$$

where each row corresponds to the unit vector for the x -, y -, and z -axis for the G expressed in F frame coordinates, respectively. Also, each column corresponds to the unit vector for the x -, y -, and z -axis for the F expressed in G frame coordinates, respectively. Each element, $C_{i,j}$, corresponds to the cosine of the angle between the i -axis of the G frame and the j -axis of the F frame.

The DCM is related explicitly to the Euler axis and rotation angle by the formulas:

$$C_{G \leftarrow F} = \begin{bmatrix} \cos \theta + \cos^2(e_x)(1 - \cos \theta) & \cos(e_x)\cos(e_y)(1 - \cos \theta) + \cos(e_z)\sin \theta & \cos(e_x)\cos(e_z)(1 - \cos \theta) \\ \cos(e_x)\cos(e_y)(1 - \cos \theta) - \cos(e_z)\sin \theta & \cos \theta + \cos^2(e_y)(1 - \cos \theta) & \cos(e_y)\cos(e_z)(1 - \cos \theta) \\ \cos(e_x)\cos(e_z)(1 - \cos \theta) + \cos(e_y)\sin \theta & \cos(e_y)\cos(e_z)(1 - \cos \theta) - \cos(e_x)\sin \theta & \cos \theta + \cos^2(e_z) \end{bmatrix} \quad (7.28)$$

$$\theta = \arccos \frac{C_{11} + C_{22} + C_{33} - 1}{2} \quad (7.29)$$

and

$$\vec{e} = \begin{bmatrix} \frac{C_{23} - C_{32}}{2\sin \theta} \\ \frac{C_{31} - C_{13}}{2\sin \theta} \\ \frac{C_{12} - C_{21}}{2\sin \theta} \end{bmatrix} \quad (7.30)$$

where the Euler axis and rotation angle can also be determined from an eigenvalue decomposition of the DCM which will have 3 eigenvalues of 1 and $\cos \theta \pm j \sin \theta$ and the eigenvector corresponding to the 1 eigenvalue will be the Euler axis.

The fourth representation uses three **basic rotation matrices** defined as

$$C_1(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix} \quad (7.31)$$

$$C_2(\theta_y) = \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \quad (7.32)$$

$$C_3(\theta_z) = \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (7.33)$$

which make use of rotations about each primary coordinate axis. Euler proved that every possible rotation in three dimensions, i.e. the rotation angle and Euler axis, can be represented by three sequential rotations which are known as the **Euler angles**. In aerospace engineering, one typically defines the Euler angles for a 3 – 1 – 3 intrinsic rotation, also known as the **classic Euler angles** or **proper Euler angles** defined by the formula:

$$\vec{v}_G = C_3(\theta_{z*})C_1(\theta_x)C_3(\theta_z)\vec{v}_F \quad (7.34)$$

or a $3 - 2 - 1$ intrinsic rotation sequence, also known as the **Tait-Bryan Euler angles**. A $3 - 2 - 1$ sequence corresponds to the formula:

$$\vec{v}_G = C_1(\theta_x)C_2(\theta_y)C_3(\theta_z)\vec{v}_F \quad (7.35)$$

where G is the frame rotating with respect to F . Here θ_z , θ_x , and θ_{z*} are the $3 - 1 - 3$ Euler angles and θ_z , θ_y , and θ_x are the $3 - 2 - 1$ Euler angles.

The DCM is related explicitly to the $3 - 1 - 3$ Euler angles by the formulas:

$$C_{G \leftarrow F} = C_3(\theta_{z*})C_1(\theta_x)C_3(\theta_z) \quad (7.36)$$

$$C_{G \leftarrow F} = \begin{bmatrix} \cos \theta_z \cos \theta_{z*} - \cos \theta_x \sin \theta_z \sin \theta_{z*} & \sin \theta_z \cos \theta_{z*} - \cos \theta_x \cos \theta_z \sin \theta_{z*} & \sin \theta_x \sin \theta_{z*} \\ -\cos \theta_z \sin \theta_{z*} - \cos \theta_x \sin \theta_z \cos \theta_{z*} & -\sin \theta_z \sin \theta_{z*} - \cos \theta_x \cos \theta_z \cos \theta_{z*} & \sin \theta_x \cos \theta_{z*} \\ \sin \theta_x \sin \theta_z & -\sin \theta_x \cos \theta_z & \cos \theta_x \end{bmatrix} \quad (7.37)$$

and

$$\begin{bmatrix} \theta_{z*} \\ \theta_x \\ \theta_z \end{bmatrix} = \begin{bmatrix} \arctan \frac{C_{13}}{C_{23}} \\ \arccos C_{33} \\ \arctan \frac{C_{31}}{-C_{32}} \end{bmatrix} \quad (7.38)$$

The DCM is related explicitly to the $3 - 2 - 1$ Euler angles by the formulas:

$$C_{G \leftarrow F} = C_1(\theta_x)C_2(\theta_y)C_3(\theta_z) \quad (7.39)$$

$$C_{G \leftarrow F} = \begin{bmatrix} \cos \theta_y \cos \theta_z & \cos \theta_y \sin \theta_z & -\sin \theta_y \\ \sin \theta_x \sin \theta_y \cos \theta_z - \cos \theta_x \sin \theta_z & \sin \theta_x \sin \theta_y \sin \theta_z + \cos \theta_x \cos \theta_z & \sin \theta_x \cos \theta_y \\ \cos \theta_x \sin \theta_y \cos \theta_z + \sin \theta_x \sin \theta_z & \cos \theta_x \sin \theta_y \sin \theta_z - \sin \theta_x \cos \theta_z & \cos \theta_x \cos \theta_y \end{bmatrix} \quad (7.40)$$

and

$$\begin{bmatrix} \theta_x \\ \theta_y \\ \theta_z \end{bmatrix} = \begin{bmatrix} \arctan \frac{C_{23}}{C_{33}} \\ -\arcsin C_{13} \\ \arctan \frac{C_{12}}{C_{11}} \end{bmatrix} \quad (7.41)$$

It should be noted that if $\theta_x = \pm 0^\circ$, then the $3 - 1 - 3$ Euler angle transformation to the DCM loses a degree-of-freedom while if $\theta_y = \pm 90^\circ$, then the $3 - 2 - 1$ Euler angle transformation to the DCM loses a degree-of-freedom. This is known as the **Euler angle ambiguity** that must be dealt with in some flight vehicle applications, e.g., switching Euler angle conventions. Euler angles will be used throughout this textbook as they encode the attitude as three parameters, i.e. the three degrees-of-freedom (3-DOF) for the rotation of a rigid body. Notably, the axis-angle and Euler parameters use four parameters and the DCM uses nine.

To derive first and second derivatives of three-dimensional vectors in rotating reference frames, one must use the **three-dimensional angular velocity tensor** as

$$[\vec{\omega}(t)]_\times = \begin{bmatrix} 0 & -\omega_z(t) & \omega_y(t) \\ \omega_z(t) & 0 & -\omega_x(t) \\ -\omega_y(t) & \omega_x(t) & 0 \end{bmatrix} \quad (7.42)$$

and the **three-dimensional angular acceleration tensor** as

$$[\vec{\alpha}(t)]_{\times} = \begin{bmatrix} 0 & -\alpha_z(t) & \alpha_y(t) \\ \alpha_z(t) & 0 & -\alpha_x(t) \\ -\alpha_y(t) & \alpha_x(t) & 0 \end{bmatrix} \quad (7.43)$$

where one can see that the upper 2×2 blocks are the two-dimensional angular velocity tensor with $\omega(t) = \omega_z(t)$ and $\alpha(t) = \alpha_z(t)$. Note also that these skew-symmetric matrices perform the cross product between two vectors, i.e.

$$\vec{\omega} \times \vec{v} = [\vec{\omega}]_{\times} \vec{v} = -[\vec{v}]_{\times} \vec{\omega} = -\vec{v} \times \vec{\omega} \quad (7.44)$$

Thus, by analogy, one has

$$\begin{aligned} \dot{C}_{F \leftarrow G} &= [\vec{\omega}_{G/F,F}]_{\times} C_{F \leftarrow G} \\ \dot{C}_{F \leftarrow F} &= -C_{F \leftarrow G} [\vec{\omega}_{G/F,F}]_{\times} \\ \dot{C}_{G \leftarrow F} &= C_{G \leftarrow F} [\vec{\omega}_{G/F,G}]_{\times} \\ \dot{C}_{G \leftarrow F} &= -[\vec{\omega}_{G/F,G}]_{\times} C_{G \leftarrow F} \end{aligned} \quad (7.45)$$

$$\vec{v}_G(t) = \vec{v}_G(t) = \dot{\vec{p}}_G(t) + [\vec{\omega}_{G/F,G}]_{\times} \vec{p}_G(t) \quad (7.46)$$

$$\vec{a}_G(t) = \ddot{\vec{v}}_G(t) + [\vec{\omega}_{G/F,G}]_{\times} \vec{v}_G(t) \quad (7.47)$$

and

$$\vec{a}_G(t) = \ddot{\vec{p}}_G(t) + 2[\vec{\omega}_{G/F,G}]_{\times} \dot{\vec{p}}_G(t) + [\vec{\alpha}_{G/F,G}]_{\times} \vec{p}_G(t) + [\vec{\omega}_{G/F,G}]_{\times} [\vec{\omega}_{G/F,G}]_{\times} \vec{p}_G(t) \quad (7.48)$$

where

$$2[\vec{\omega}_{G/F,G}]_{\times} \dot{\vec{p}}_G(t) \quad (7.49)$$

is the **three-dimensional Coriolis acceleration**,

$$[\vec{\alpha}_{G/F,G}]_{\times} \vec{p}_G(t) \quad (7.50)$$

is the **three-dimensional Euler acceleration**, and

$$[\vec{\omega}_{G/F,G}]_{\times} [\vec{\omega}_{G/F,G}]_{\times} \vec{p}_G(t) \quad (7.51)$$

is the **three-dimensional centrifugal acceleration**.

Lastly, by definition, the basic rotation matrices have derivatives

$$\dot{C}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\sin \theta_x & \cos \theta_x \\ 0 & -\cos \theta_x & -\sin \theta_x \end{bmatrix} \dot{\theta}_x = - \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta_x & \sin \theta_x \\ 0 & -\sin \theta_x & \cos \theta_x \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\dot{\theta}_x \\ 0 & \dot{\theta}_x & 0 \end{bmatrix} = -C_1 \begin{bmatrix} \dot{\theta}_x \\ 0 \\ 0 \end{bmatrix}_{\times} \quad (7.52)$$

$$\dot{C}_2 = \begin{bmatrix} -\sin \theta_y & 0 & \cos \theta_y \\ 0 & 0 & 0 \\ -\cos \theta_y & 0 & -\sin \theta_y \end{bmatrix} \dot{\theta}_y = - \begin{bmatrix} \cos \theta_y & 0 & -\sin \theta_y \\ 0 & 1 & 0 \\ \sin \theta_y & 0 & \cos \theta_y \end{bmatrix} \begin{bmatrix} 0 & 0 & \dot{\theta}_y \\ 0 & 0 & 0 \\ -\dot{\theta}_y & 0 & 0 \end{bmatrix} = -C_2 \begin{bmatrix} 0 \\ \dot{\theta}_y \\ 0 \end{bmatrix}_{\times} \quad (7.53)$$

$$\dot{C}_3 = \begin{bmatrix} -\sin \theta_z & \cos \theta_z & 0 \\ -\cos \theta_z & -\sin \theta_z & 0 \\ 0 & 0 & 0 \end{bmatrix} \dot{\theta}_z = - \begin{bmatrix} \cos \theta_z & \sin \theta_z & 0 \\ -\sin \theta_z & \cos \theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -\dot{\theta}_z & 0 \\ \dot{\theta}_z & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} = -C_3 \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_z \end{bmatrix}_{\times} \quad (7.54)$$

Thus, by the product rule and the definition of the intermediate axes of rotation, the derivatives of the 3–1–3 Euler angles are related explicitly to the angular velocity by the

$$\vec{\omega}_{G/F,G} = C_3(\theta_{z*}) \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_{z*} \end{bmatrix} + C_3(\theta_{z*})C_1(\theta_x) \begin{bmatrix} \dot{\theta}_x \\ 0 \\ 0 \end{bmatrix} + C_3(\theta_{z*})C_1(\theta_x)C_3(\theta_z) \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_z \end{bmatrix} \quad (7.55)$$

which result in the following formulas:

$$\vec{\omega}_{G/F,G} = \begin{bmatrix} \sin \theta_x \sin \theta_z & \cos \theta_z & 0 \\ \sin \theta_x \cos \theta_z & -\sin \theta_z & 0 \\ \cos \theta_x & 0 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_{z*} \\ \dot{\theta}_x \\ \dot{\theta}_z \end{bmatrix} \quad (7.56)$$

and

$$\begin{bmatrix} \dot{\theta}_{z*} \\ \dot{\theta}_x \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} \sin \theta_z \csc \theta_x & \cos \theta_z \csc \theta_x & 0 \\ \cos \theta_z & -\sin \theta_z & 0 \\ -\sin \theta_z \cot \theta_x & -\cos \theta_z \cot \theta_x & 1 \end{bmatrix} \vec{\omega}_{G/F,G} \quad (7.57)$$

while the derivatives of the 3–2–1 Euler angles are related explicitly to the angular velocity by

$$\vec{\omega}_{G/F,G} = C_1(\theta_x) \begin{bmatrix} \dot{\theta}_x \\ 0 \\ 0 \end{bmatrix} + C_1(\theta_x)C_2(\theta_y) \begin{bmatrix} 0 \\ \dot{\theta}_y \\ 0 \end{bmatrix} + C_1(\theta_x)C_2(\theta_y)C_3(\theta_z) \begin{bmatrix} 0 \\ 0 \\ \dot{\theta}_z \end{bmatrix} \quad (7.58)$$

which results in

$$\vec{\omega}_{G/F,G} = \begin{bmatrix} 1 & 0 & -\sin \theta_x \\ 0 & \cos \theta_x & \sin \theta_x \cos \theta_y \\ 0 & -\sin \theta_x & \cos \theta_x \cos \theta_y \end{bmatrix} \begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} \quad (7.59)$$

and can be solved for the inverse conversion as

$$\begin{bmatrix} \dot{\theta}_x \\ \dot{\theta}_y \\ \dot{\theta}_z \end{bmatrix} = \begin{bmatrix} 1 & \sin \theta_x \tan \theta_y & \cos \phi \tan \theta_y \\ 0 & \cos \theta_x & -\sin \theta_x \\ 0 & \sin \theta_x \sec \theta_y & \cos \theta_x \sec \theta_y \end{bmatrix} \vec{\omega}_{G/F,G} \quad (7.60)$$

where it should be noted that these conversion matrices are not transposes of each other as for DCMs.

References

For more information, please refer to the following

- Nelson, R. C., “3.3 Orientation and Position of the Airplane,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 101-103

- Schmidt, D. K., “1.3 Vectors, Coordinate Transformations, and Direction-Cosine Matrices,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 4-10
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “1.3 Matrix Operations on Vector Coordinates,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 8-15
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “1.4 Rotational Kinematics,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 16-20

7.3 Aerospace Vehicle Reference Frames and Euler Angles

Celestial and Earth-Centered Frames

Flight vehicles operate either in or outside of the atmospheres of celestial bodies which have their own gravitational motion. Although general relativity implies that there are no true inertial frames around gravitating bodies, the use of the **International Celestial Reference Frame (ICRF)** allows one to define the positions of astronomical objects as the ICRF is based on hundreds of extra-galactic radio sources, mostly quasars, distributed around the entire sky. The ICRF does not exhibit any measurable angular rotation since the extragalactic sources used to define the ICRF are so far away and appear stationary to any sensing technology. However, their positions can be measured very accurately by **Very Long Baseline Interferometry (VLBI)**. The positions of most are known to 1 milliarcsecond or better. Its origin is at the barycenter of the Solar System, with axes that are intended to “show no global rotation with respect to a set of distant extragalactic objects.”

In flight dynamics, one often desires to understand a vehicle’s motion relative to the celestial body whose gravitational force is forcing an orbit or resisting flight on the vehicle. As Earth is the most orbited celestial body, this textbook considers only **Earth-centered frames** for celestial body frames which can be of two different types.

The first is an **Earth-centered, inertial (ECI) frame** (subscript *I*) which is useful for spacecraft orbiting around Earth where the Sun’s gravity can be negligible and high fidelity dynamic models of aircraft traveling long distances. This frame is defined as

- The origin is the Earth’s center of mass.
- The x_I -, y_I -, and z_I -axes are the same as the ICRF.

The second is an **Earth-centered, Earth-fixed (ECEF) frame** (subscript *E*) is attached to the Earth’s surface, thereby it rotates with the Earth’s surface, i.e. the coordinates of a point on the surface of the Earth do not change. This frame is defined as

- The origin is the Earth’s center of mass.
- The x_E -axis passes through intersection of the prime meridian and the equator, located just south of west Africa.

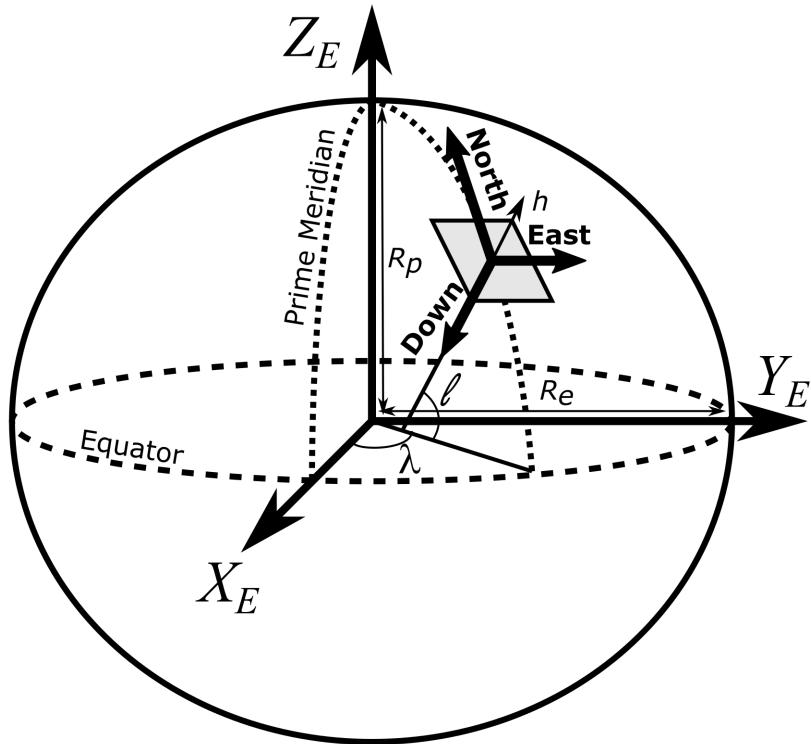
- The y_E -axis is orthogonal to both x_E - and z_E -axes according to right-hand-rule located just south of India along the equator.
- The z_E -axis passes through true north which does not coincide with Earth's instantaneous rotational axis because of the Earth's wobble, but its average.

However, it should be noted that the surface of the Earth is not perfectly spherical. It is actually not a fully geometrically realizable shape at all and its surface and center of mass changes with time. Thus, the discipline of **geodesy** has developed which is the study of Earth's shape. This science typically uses two different approximations for the Earth's shape, the geoid and the reference ellipsoid. The **geoid** which is an idealized equilibrium surface of the Earth's gravitational potential which varies according to its crust formation and is also known as the **International Terrestrial Reference Frame (ITRF)**.

A **reference ellipsoid** approximates the geoid and has an equatorial radius, R_e , longer than its polar radius, R_p , a shape also known as an **oblate spheroid**. A common model for the reference ellipsoid is the **World Geodetic System (WGS)**. The traditional coordinates used for the reference ellipsoid are **latitude, longitude, and altitude, (LLA)**, also known as **geodetic coordinates**. The **prime meridian** is defined as 0° longitude. The **equator** is defined as 0° latitude. **Mean sea level (MSL)** is defined as 0 altitude and can be considered the ideal continuous surface of the ocean in the absence of currents and air pressure variations and whose surface continues under the continental masses. Geodetic coordinates are ellipsoidal coordinates, thus these angles are defined according to ellipsoidal trigonometry. Thus, because these are ellipsoidal as opposed to spherical, the latitude angle does not define the angle between the Earth's center and a point on the surface and altitude does not always intersect the Earth's center of mass, but both have an analytical offset.

An important aspect of the ECEF reference ellipsoid is the description of the **local tangent plane (LTP)** at a reference latitude, ℓ_0 , longitude, λ_0 , and altitude, h_0 . Common axes for that can be based on the LTP include the east-west direction, i.e. tangent to the latitude parallels, the north-south direction, i.e. tangent to the longitudinal meridians, and the up-down direction, i.e. normal to the reference ellipsoid.

The relationship between the ECEF frame, LLA, and the north-east-down directions of the LTP can be visualized through the following figure.



Vehicle-Centered Frames

The **vehicle-centered inertial frame** (subscript I^*) is typically used for describing the attitude of a vehicle relative to the ICRF or ECI frame. This frame is defined as follows:

- The origin is the vehicle's center of mass.
- x_{I^*} -, y_{I^*} -, and z_{I^*} -axes are the same as the ICRF or obtained from a constant DCM transformation from the ICRF axes.

The **navigation frame** (subscript N) is typically used for the describing the long-term motion of the vehicle relative to the Earth's surface as represented by the reference ellipsoid. This frame is defined as follows:

- The origin is the vehicle's center of mass.
- x_N - and y_N -axes: aligned with east-west or north-south
- z_N -axis: aligned with up-down direction, i.e., orthogonal to the LTP

Thus, the navigation frame is typically defined as either an **East-North-Up (ENU)** or a **North-East-Down (NED)** navigation frame. NED is traditionally used for aerospace vehicles as objects of interest are typically below aerospace vehicles.

The **geodetic frame** (subscript G) is an alternative to the navigation frame and is often used for orbiting satellites. This frame is defined as

- The origin is the vehicle's center of mass.
- x_G -axis is orthogonal to y_G - and z_G -axes forming right-handed coordinate frame.
- y_G -axis is directed along the angular momentum vector of the vehicle, e.g. orthogonal to the orbital plane.
- z_G -axis is directed along the Down direction of instantaneous LTP, i.e., orthogonal to the LTP.

The nadir-pointing **local-vertical, local-horizontal (LVLH) frame** (subscript L). This frame is defined as follows:

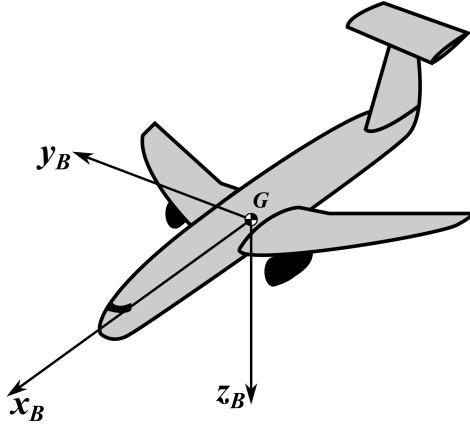
- The origin is the vehicle's center of mass.
- The x_L -axis, also known as the **local-horizontal axis**, is orthogonal to x_L - and z_L -axes forming a right-handed coordinate frame.
- The y_L -axis is directed along the angular momentum vector of the vehicle, e.g. orthogonal to the orbital plane.
- The z_L -axis, also known as the **local-vertical axis**, is directed towards the center of the Earth.

Notably, the z_L -axis may be slightly misaligned with the z_N - and z_G -axes because the down direction for the LTP only intersect the Earth's center of mass along the Equator and at the poles.

The **body-fixed frame** (subscript B) is attached to the rigid body of the aerospace vehicle, thus it is ideal for geometric configuration and structural modeling. This frame is defined as follows:

- The origin is the aerospace vehicle's center of mass.
- The x_B axis points out the front of the aerospace vehicle, typically “along” the nominal path of travel, also known as the **longitudinal axis**.
- The y_B axis points out the right side of the aerospace vehicle, also known as the **lateral axis**.
- The z_B axis points is orthogonal to x_B - and z_B -axes forming a right-handed coordinate frame.

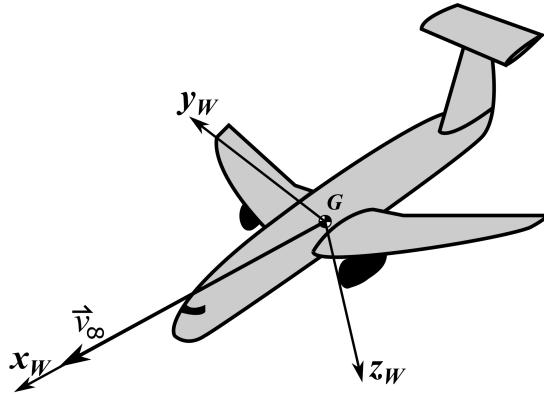
This frame is depicted for an airplane in the following figure.



The **wind frame** (subscript W) relates the **free-stream airflow** that an aircraft encounters as it flies, thus it is ideal for aerodynamics modeling and is particular to aircraft dynamics and control. The frame is defined as

- The origin is the aircraft's center of gravity G .
- The x_W axis is colinear with free-stream velocity, \vec{v}_∞ .
- The z_W axis is in the plane of symmetry of the aircraft, positive below the aircraft.
- The y_W axis is orthogonal to both.

This frame is depicted for an airplane in the following figure.

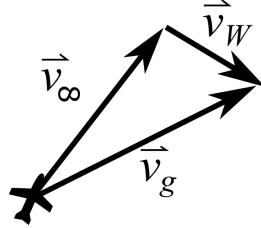


Here the aircraft velocity \vec{v}_∞ is the velocity relative to the air mass that the aircraft is traveling through, i.e. the **free-stream velocity** and v_∞ is the **airspeed**. This is different than the velocity

of the aircraft with respect to the planet's surface, i.e. the **ground speed** vector \vec{v}_g . These velocity vectors are related through the **wind speed** vector, \vec{v}_w , as

$$\vec{v}_g = \vec{v}_\infty + \vec{v}_w \quad (7.61)$$

an equation also known as the **wind triangle** as this vector relationship forms a triangle, i.e.



Note \vec{v}_w is the velocity of the air mass relative to the ground.

Euler Angles Relating Vehicle-Centered Frames

The $3 - 1 - 3$ Euler angles for describing the body-fixed frame relative to the vehicle-centered, inertial frame are the **precession angle**, ϕ_p , **nutation angle**, θ_n , and **spin angle**, ψ_s , where the subscripts have been added to differentiate these Euler angles from the $3 - 2 - 1$ navigation-to-body frame Euler angles, roll, pitch, and yaw. Thus, a vector expressed in vehicle-centered inertial frame coordinates, \vec{v}_{I^*} , can be expressed as a vector in body-fixed frame coordinates, \vec{v}_B , through the sequence

$$\vec{v}_B = C_3(\psi_s)C_1(\theta_n)C_3(\phi_p)\vec{v}_{I^*} \quad (7.62)$$

$$\vec{v}_B = C_{B \leftarrow I^*}\vec{v}_{I^*} \quad (7.63)$$

where

$$C_{B \leftarrow I^*} = \begin{bmatrix} \cos \phi_p \cos \psi_s - \cos \theta_n \sin \phi_p \sin \psi_s & \sin \phi_p \cos \psi_s - \cos \theta_n \cos \phi_p \sin \psi_s & \sin \theta_n \sin \psi_s \\ -\cos \phi_p \sin \psi_s - \cos \theta_n \sin \phi_p \cos \psi_s & -\sin \phi_p \sin \psi_s - \cos \theta_n \cos \phi_p \cos \psi_s & \sin \theta_n \cos \psi_s \\ \sin \theta_n \sin \psi_s & -\sin \theta_n \cos \psi_s & \cos \theta_n \end{bmatrix} \quad (7.64)$$

The $3 - 2 - 1$ Euler angles for describing the wind frame relative to the navigation frame are the **bank angle**, μ , **flight-path angle**, γ , and **heading angle**, σ . Thus, a vector expressed in navigation frame coordinates, \vec{v}_N , can be expressed as a vector in wind frame coordinates, \vec{v}_W , through the sequence

$$\vec{v}_W = C_1(\mu)C_2(\gamma)C_3(\sigma)\vec{v}_N \quad (7.65)$$

$$\vec{v}_W = C_{W \leftarrow N}\vec{v}_N \quad (7.66)$$

where

$$C_{W \leftarrow N} = \begin{bmatrix} \cos \gamma \cos \sigma & \cos \gamma \sin \sigma & -\sin \gamma \\ \sin \mu \sin \gamma \cos \sigma - \cos \mu \sin \sigma & \sin \mu \sin \gamma \sin \sigma + \cos \mu \cos \sigma & \sin \mu \cos \gamma \\ \cos \mu \sin \gamma \cos \sigma + \sin \mu \sin \sigma & \cos \mu \sin \gamma \sin \sigma - \sin \mu \cos \sigma & \cos \mu \cos \gamma \end{bmatrix} \quad (7.67)$$

It should be noted that the heading angle can be arbitrarily set as subsequent rotations in heading can be performed before the other navigation-to-body-fixed frame Euler angles, but is typically referenced to north.

The 3 – 2 – 1 Euler angles for describing the body-fixed frame relative to the navigation or LVLH frame are the **roll angle**, ϕ , **pitch angle**, θ , and **yaw angle**, ψ . Thus, a vector expressed in navigation frame coordinates, \vec{v}_N , can be expressed as a vector in body-fixed frame coordinates, \vec{v}_B , through the sequence

$$\vec{v}_B = C_1(\phi)C_2(\theta)C_3(\psi)\vec{v}_{N \text{ or } L} \quad (7.68)$$

$$\vec{v}_B = C_{B \leftarrow N \text{ or } L}\vec{v}_{N \text{ or } L} \quad (7.69)$$

where

$$C_{B \leftarrow N \text{ or } L} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \quad (7.70)$$

These Euler angles are used to represent the orientation or **attitude** of the aircraft as they provide a description of how the rigid body of the aircraft is currently oriented in 3D space. It should be noted that the yaw angle can be arbitrarily set as subsequent rotations in yaw can be performed before the other angles.

The Euler angles for describing the body-fixed frame relative to the wind frame are the **angle of attack**, α , and the **sideslip angle**, β . Thus, a vector expressed in wind frame coordinates, \vec{v}_W , can be expressed as a vector in body-fixed frame coordinates, \vec{v}_B , through the sequence

$$\vec{v}_B = C_2(\alpha)C_3(-\beta)\vec{v}_W \quad (7.71)$$

$$\vec{v}_B = C_{B \leftarrow W}\vec{v}_W \quad (7.72)$$

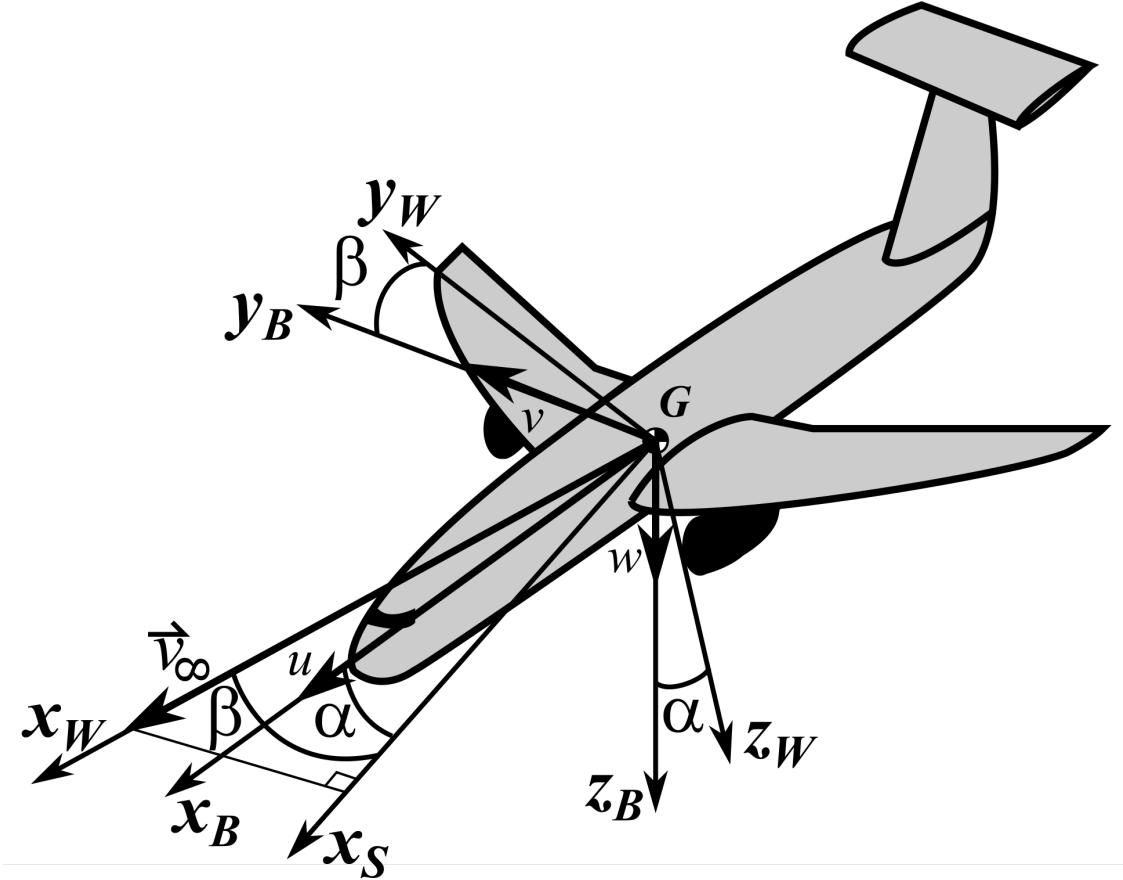
where

$$\vec{v}_B = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \vec{v}_W \quad (7.73)$$

Of particular importance for this rotation is the representation of the airspeed vector \vec{v}_∞ with magnitude v_∞ expressed in the body-fixed frame as the components

$$\vec{v}_{\infty, B} = [u \ v \ w]^T \quad (7.74)$$

which can be visualized in the following diagram



where x_S is the x -axis of the **stability frame** which is defined later.

From the previous equations, one has

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} v_\infty \\ 0 \\ 0 \end{bmatrix} \quad (7.75)$$

$$\begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} v_\infty \cos \alpha \cos \beta \\ v_\infty \sin \beta \\ v_\infty \sin \alpha \cos \beta \end{bmatrix} \quad (7.76)$$

Dividing the first row by the third row, one has

$$\frac{u}{w} = \frac{v_\infty \cos \alpha \cos \beta}{v_\infty \sin \alpha \cos \beta} \quad (7.77)$$

$$\frac{u}{w} = \frac{\cos \alpha}{\sin \alpha} \quad (7.78)$$

$$\frac{u}{w} = \frac{1}{\tan \alpha} \quad (7.79)$$

which provides the relationship between the angle of attack and the components of the airspeed vector as

$$\alpha = \tan^{-1} \frac{w}{u} \quad (7.80)$$

While isolating the second row, one has

$$v = v_\infty \sin \beta \quad (7.81)$$

which provides the relationship between the sideslip angle and the components of the airspeed vector as

$$\beta = \sin^{-1} \frac{v}{v_\infty} \quad (7.82)$$

Moreover, it should be pointed out that there is an implicit relationship between these sets of Euler angles, e.g.

$$C_{W \leftarrow N} = C_{W \leftarrow B} C_{B \leftarrow N} \quad (7.83)$$

which, in general, results in complicated trigonometric equations.

However, for linearized flight dynamics, it is useful to form the following approximate relationships between the vehicle frame Euler angles using the small angle approximations for sin and cos. For Equation 7.83, this becomes

$$\begin{bmatrix} 1 & \sigma & -\gamma \\ \mu\gamma - \sigma & \mu\gamma + 1 & \mu \\ \gamma + \mu\sigma & \gamma\mu - \mu & 1 \end{bmatrix} \approx \begin{bmatrix} 1 & \beta & \alpha \\ -\beta & 1 & 0 \\ -\alpha & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & \psi & -\theta \\ \phi\theta - \psi & \phi\theta + 1 & \phi \\ \theta + \phi\psi & \theta\psi - \phi & 1 \end{bmatrix} \quad (7.84)$$

By computing the element in the second row and third column, one has

$$\mu \approx \theta\beta + \phi \quad (7.85)$$

and discarding higher-order terms, one has

$$\mu \approx \phi \quad (7.86)$$

By computing the element in the first row and third column

$$-\gamma \approx -\theta + \beta\phi + \alpha \quad (7.87)$$

and discarding higher-order terms, one has

$$\gamma \approx \theta - \alpha \quad (7.88)$$

By computing the element in the first row and second column, one has

$$\sigma \approx \psi + \beta + \phi\theta(1 + \beta) - \psi\theta^2 \quad (7.89)$$

and discarding higher-order terms, one has

$$\sigma \approx \psi + \beta \quad (7.90)$$

It should be noted that small σ and ψ here correspond to small *changes* in the *nominal* heading and yaw as these both can be set to an arbitrary reference direction in the rotation sequence.

References

For more information, please refer to the following

- Curtis, H. D., “9.9 Euler Angles,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 448-459
- Curtis, H. D., “9.10 Yaw, Pitch, and Roll Angles,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 459-463
- Nelson, R. C., “3.3 Orientation and Position of the Airplane,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 101-103
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “1.4 Rotational Kinematics,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 16-20
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “1.6 Geodesy, Coordinate Systems, Gravity,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 23-34

7.4 Steady-Flight Conditions for Aerospace Vehicles

When modeling the motion of physical objects using classical mechanics, one can choose between three different models. First, a **point-mass model** defines an object only by its mass. Thus, the point-mass model assumes that an object’s motion is completely characterized by the motion of its center of mass. Second, a **rigid-body model** defines an object by a static distribution of mass in space. Thus, the rigid body model assumes that an object’s motion is completely characterized by the motion of its center of mass and the rotation of its structure. Third, a **deformable-body model** defines an object by a dynamic distribution of mass in space that experiences deformations caused by strain. A sub-type of the deformable body model is the **elastic-body model** which assumes the deformations caused by strain are completely recoverable. Thus, the elastic body model assumes that an object’s motion is completely characterized by the motion of its nominal center of mass, the rotation of nominal structure, and the vibration of body about its nominal structure where, notably, these vibrations are known as the structural modes of the object. In this introductory part of the textbook focuses on the rigid body model for aerospace vehicles which is “good enough” for the majority of aerospace vehicles, as their structures are designed so that the structural modes occur at much higher frequencies than the aerodynamic or astrodynamical modes. However, modern aircraft often incorporate flexible structures which necessitate the elastic body model, i.e. aeroelastic dynamics. This elastic aerospace vehicle model is considered in a later section of this textbook.

This section looks at the point-mass models for airplanes and satellites and the analysis of these dynamics models for computing steady-flight conditions. The point-mass model for airplanes allows for more direct investigation of the airplane’s ability to follow a commanded **flight profile** which can be defined as a commanded velocity vector, often specified as a forward speed, rate-of-climb, and heading, similar to air traffic control commands. Likewise, the point-mass model for a satellite allows for more direct investigation of a satellite’s ability to follow a commanded **reference trajectory** which can be defined as a series of predefined orbits.

Force Modeling

In Newtonian mechanics, point-mass dynamics are represented using the **Newton's second law** governing translation which, in modern form, relates the forces acting on the point-mass to the linear momentum. This results in the vector-valued differential equation

$$\sum \vec{F} = \frac{d}{dt} \vec{p} = \frac{d}{dt} (m \vec{v}) \quad (7.91)$$

where $\sum \vec{F}$ is the total force on the rigid body, \vec{p}_G is the linear momentum, m is the mass, and \vec{v} is the velocity of the point-mass. This equation forms the basis of the three degrees-of-freedom (DOF) equations of motion for point-mass aerospace vehicles where the three DOF are the velocity components of the vehicle.

Fundamentally, aerospace vehicles always encounter gravitational forces. In Newtonian mechanics, **Newton's law of gravitation** states that the **gravitational force**, F_g , also known as the **weight**, W , is a function of the distance r between the centers of mass for the vehicle m and a celestial body M , i.e.

$$F_g = W = \frac{GMm}{r^2} = \frac{\mu m}{r^2} \quad (7.92)$$

where G is the gravitational constant, i.e. $6.674 \times 10^{-11} \text{ m}^3 \text{kg}^{-1} \text{s}^{-2}$ and $\mu = GM$ is the **standard gravitational parameter** of the celestial body. It should be noted that though a celestial body's gravity field typically also contains variation which is neglected for airborne vehicles while it will be introduced as a disturbance in the analysis of spacecraft dynamics and control in this textbook.

For aerospace vehicles “near” Earth, one can model the gravitational force from Earth only as

$$F_g = mg \quad (7.93)$$

where g is Earth's **gravitational acceleration**. It should be noted that g generally varies as a function of latitude, altitude, and local topography and geology. This part of the textbook will assume a **spherical-Earth gravity model** for the gravitational acceleration as a function of altitude, h , by the equation

$$g(h) = g_0 \left(\frac{\bar{R}_E}{\bar{R}_E + h} \right)^2 \quad (7.94)$$

where \bar{R}_E is the **Earth's mean radius**, i.e. 6.3710088×10^6 m, and g_0 is the **standard gravitational acceleration**, i.e. 9.80665 m/s 2 . Higher fidelity modeling of the gravitational force is discussed in later parts of this textbook.

To overcome gravity and obtain sustained flight, aerospace vehicles must have propulsion systems which produce the propulsive forces and moments. These systems typically use either rotating propellers, rotors, and/or reaction engines, e.g. jet, rocket, or ion, which are affixed to the vehicle's structure. Thus, one typically defines the **propulsive force** vector, \vec{F}_p , and the **propulsive moment** vector, \vec{M}_p , in the body frame as a function of **thrust** vectors, \vec{T} . The propulsion system is primarily used to translate the entire vehicle; however, some propulsion systems are used to steer aerospace vehicles, a design known as **thrust vectoring**. This textbook will not consider propulsion system modeling in detail, but will assume that one can define a dynamical system model for the propulsion that may be dependent on the flight conditions, e.g. airspeed and air density, and can then be approximated by first- or second-order dynamics, similar to other

actuators. It should also be noted that rotating propellers and rotors create a changing mass distribution due to their rotation; however, this can be accommodated in the rigid aerospace vehicle equations of motion and is addressed in a later part of the textbook.

The aerodynamic forces and moments are due to the air pressure distribution around the aerospace vehicle. The **aerodynamic force** vector, \vec{F}_a , resist/assist translation through the air mass while the **aerodynamic moment** vector, \vec{M}_a , cause the aircraft to rotate due to the varying pressure distribution over the airborne body. The aerodynamic forces and moments are the primary factor in airborne vehicles and are often negligible for spaceborne vehicles, except at some low planetary orbits. These forces and moments for fixed-wing aircraft are introduced as conventional analytical models in a later section of this chapter using basic aerodynamic theory. It should be noted that the determination of these models for general aircraft is studied as **aircraft system identification (SID)** which typically employs **optimal parameter estimation** and is discussed in later parts of this textbook.

The solar forces and moments are due to the solar radiation pressure distribution around the aerospace vehicle. The **solar force** vector, \vec{F}_s , resists translation through space while the **solar moment** vector, \vec{M}_s , causes the spacecraft to rotate due to the varying pressure distribution over the spacecraft. The solar forces and moments are a factor in spacecraft and are negligible for airborne vehicles. These forces and moments are introduced as disturbances in the analysis of spacecraft dynamics and control in this textbook.

Point-Mass Airplane Steady-Flight

Notably, this and the next introductory sections of the textbook on aircraft will often use the **flat-Earth approximation** which assumes that the entire Earth's surface can be approximated by a single coordinate frame which can be considered as the “average” LTP in the area of operation for aircraft. Furthermore, in this case, the navigation frame coordinate axes remain static, i.e. “inertial,” which allows one to more easily compute the dynamics for aircraft, and the “down” direction aligns with the gravitational force, mg . This approximation is typically “good enough” for aircraft dynamics and control analysis and design. Although with hypersonic aircraft velocities and/or long-distance flight analysis, the flat-Earth approximation may not be suitable even for this case. The additional effects of non-flat-Earth modeling on flight dynamics is presented in an advanced section of this textbook.

Fixed-wing aircraft generate aerodynamic forces from several different lifting surfaces for which one must define the aerodynamic forces for the entire aircraft. Similar to finite-wing theory (see appendix A.1), one can model a fixed-wing aircraft’s total aerodynamic force vector, \vec{F}_a , in the wind frame as

$$\vec{F}_{a,W} = \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \quad (7.95)$$

where D is the **drag force** for the entire fixed-wing aircraft, S is the **side force** for the entire fixed-wing aircraft, and L is the **lift force** for the entire fixed-wing aircraft.

Here, one typically models each of these by their vehicle coefficients, i.e.

$$L = Q_\infty S_w C_L \quad (7.96)$$

$$S = Q_\infty S_w C_S \quad (7.97)$$

$$D = Q_\infty S_w C_D \quad (7.98)$$

where, one can model the drag coefficient as the sum of the parasitic drag, induced drag, and potentially wave drag, i.e.,

$$C_D = C_{D,p} + C_{D,i} + C_{D,w} = C_{D,0} + k C_L^2 \quad (7.99)$$

For subsonic flight, $C_{D,w} = 0$, $C_{D,0} = C_{D,p}$, and

$$k = k_w = \frac{1}{\pi e_{eff} AR_w} \quad (7.100)$$

where e_{eff} is the **effective Oswald's efficiency** which can depend on the Mach number and C_L near and above the drag-divergence Mach number. For supersonic flight,

$$C_{D,0} = C_{D,p} + \frac{49(t/\bar{c})^2}{\sqrt{\mathcal{M}^2 - 1}} \quad (7.101)$$

$$k = k_i + k_w = \frac{1}{\pi e_{eff} AR_w} + \frac{AR_w(\mathcal{M}^2 - 1)}{4AR_w\sqrt{\mathcal{M}^2 - 1} - 2} \cos \Lambda_{LE} \quad (7.102)$$

while in the transonic region past $\mathcal{M} = 1$, k transitions slowly from k_i to $k_i + k_w$.

Notably, these wind frame aerodynamic forces can be rotated to the body-fixed frame by

$$\vec{F}_{a,B} = C_{B \leftarrow W} \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \quad (7.103)$$

$$\vec{F}_{a,B} = \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \quad (7.104)$$

or

$$\vec{F}_{a,B} = \begin{bmatrix} -D \cos \alpha \cos \beta - S \cos \alpha \sin \beta + L \sin \alpha \\ S \cos \beta - D \sin \beta \\ -D \sin \alpha \cos \beta - S \sin \alpha \sin \beta - L \cos \alpha \end{bmatrix} \quad (7.105)$$

In addition, S here without a subscript should not be confused with the surface area of a lifting surface which always has a specifying subscript with it in this textbook.

For airplanes, one typically models the propulsive force as

$$\vec{F}_{p,B} = \begin{bmatrix} T \cos \theta_T \\ 0 \\ T \sin \theta_T \end{bmatrix} \quad (7.106)$$

where T is the **thrust force** and θ_T is a potential offset angle with respect to the x_B -axis of the body-fixed frame. Moreover, a propulsive moment may also be present nominally about the y_B -axis as

$$\vec{M}_{p,B} = \begin{bmatrix} 0 \\ T(z_T \cos \theta_T - x_T \sin \theta_T) \\ 0 \end{bmatrix} \quad (7.107)$$

where (x_T, z_T) denotes the location of the thrust force in the $x_B - z_B$ plane. It should be noted that often $\theta_T = 0^\circ$ or approximately and T is generally a function of the airspeed, altitude, and throttle setting.

For this point-mass model, one can use the wind frame for the translation equation, i.e.

$$\sum \vec{F}_W = \frac{d}{dt}(m \vec{v}_W) = m(\dot{\vec{v}}_W + \vec{\omega}_{W/N} \times \vec{v}_W) \quad (7.108)$$

Next, defining the angular velocity of the wind frame relative to the navigation as

$$\vec{\omega}_{W/N} = [p_W \quad q_W \quad r_W]^T \quad (7.109)$$

and recalling, by definition

$$\vec{v}_W = \begin{bmatrix} v_\infty \\ 0 \\ 0 \end{bmatrix} \quad (7.110)$$

one has

$$\vec{F}_{a,W} + \vec{F}_{p,W} + \vec{F}_{g,W} = \begin{bmatrix} m\dot{v}_\infty \\ 0 \\ 0 \end{bmatrix} + m \begin{bmatrix} p_W \\ q_W \\ r_W \end{bmatrix} \times \begin{bmatrix} v_\infty \\ 0 \\ 0 \end{bmatrix} \quad (7.111)$$

Furthermore, one can model the gravitational force as

$$\vec{F}_{g,W} = C_{W \leftarrow N}(\mu, \gamma, \sigma) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (7.112)$$

the propulsive force assuming that $\theta_T = 0$ as

$$\vec{F}_{p,W} = C_{W \leftarrow B}(\alpha, \beta) \begin{bmatrix} T \\ 0 \\ 0 \end{bmatrix} \quad (7.113)$$

and the aerodynamic force as

$$\vec{F}_{a,W} = \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \quad (7.114)$$

Thus, one has

$$\begin{bmatrix} -D \\ S \\ -L \end{bmatrix} + \begin{bmatrix} T \cos \alpha \cos \beta \\ -T \cos \alpha \sin \beta \\ -T \sin \alpha \end{bmatrix} + \begin{bmatrix} -mg \sin \gamma \\ mg \sin \mu \cos \gamma \\ mg \cos \mu \cos \gamma \end{bmatrix} = \begin{bmatrix} \dot{v}_\infty \\ mv_\infty r_W \\ -mv_\infty q_W \end{bmatrix} \quad (7.115)$$

Then, substituting for the wind frame angular velocity components with the navigation-to-wind frame Euler angles, i.e.

$$\begin{bmatrix} p_W \\ q_W \\ r_W \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \gamma \\ 0 & \cos \mu & \sin \mu \cos \gamma \\ 0 & -\sin \mu & \cos \mu \cos \gamma \end{bmatrix} \begin{bmatrix} \dot{\mu} \\ \dot{\gamma} \\ \dot{\sigma} \end{bmatrix} \quad (7.116)$$

Rearranging, one has

$$\begin{bmatrix} -D + T \cos \alpha \cos \beta - mg \sin \gamma \\ S - T \cos \alpha \sin \beta + mg \sin \mu \cos \gamma \\ L + T \sin \alpha - mg \cos \mu \cos \gamma \end{bmatrix} = \begin{bmatrix} m\dot{v}_\infty \\ mv_\infty (\dot{\sigma} \cos \mu \cos \gamma - \dot{\gamma} \sin \mu) \\ v_\infty (\dot{\gamma} \cos \mu + \dot{\sigma} \sin \mu \cos \gamma) \end{bmatrix} \quad (7.117)$$

where one can add control inputs to this model through the thrust T and the lift L . Lastly, one can also include the navigation frame velocity components as

$$\begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{h} \end{bmatrix} = \begin{bmatrix} v_\infty \cos \gamma \cos \sigma \\ v_\infty \cos \gamma \sin \sigma \\ v_\infty \sin \gamma \end{bmatrix} \quad (7.118)$$

where $-\dot{h} = \dot{z}_N$ is the altitude rate instead of the down rate.

With these dynamics, one can analyze the **point-mass equilibrium flight conditions**, also known as the **point-mass steady-flight conditions** which dictates $\dot{v}_\infty = \dot{\gamma} = 0$. Then, the forces and trim states of the airplane can be related as

$$\begin{bmatrix} -\bar{D} + \bar{T} \cos \bar{\alpha} \cos \bar{\beta} - mg \sin \bar{\gamma} \\ \bar{S} - \bar{T} \cos \bar{\alpha} \sin \bar{\beta} + mg \sin \bar{\mu} \cos \bar{\gamma} \\ -\bar{T} \sin \bar{\alpha} - \bar{L} + mg \cos \bar{\mu} \cos \bar{\gamma} \end{bmatrix} = \begin{bmatrix} 0 \\ m\bar{v}_\infty \dot{\sigma} \cos \bar{\mu} \cos \bar{\gamma} \\ m\bar{v}_\infty \dot{\sigma} \sin \bar{\mu} \cos \bar{\gamma} \end{bmatrix} \quad (7.119)$$

Of particular note is the **coordinated steady-flight condition** when $\bar{S} = 0$ and $\bar{\beta} = 0^\circ$. Then, by substitution, one has

$$\begin{bmatrix} -\bar{D} + \bar{T} \cos \bar{\alpha} - mg \sin \bar{\gamma} \\ mg \sin \bar{\mu} \cos \bar{\gamma} \\ -\bar{T} \sin \bar{\alpha} - \bar{L} + mg \cos \bar{\mu} \cos \bar{\gamma} \end{bmatrix} = \begin{bmatrix} 0 \\ m\bar{v}_\infty \dot{\sigma} \cos \bar{\mu} \cos \bar{\gamma} \\ m\bar{v}_\infty \dot{\sigma} \sin \bar{\mu} \cos \bar{\gamma} \end{bmatrix} \quad (7.120)$$

Finally, defining the **instantaneous radius of curvature** in the navigation frame, R_c , as

$$R_c = \frac{\bar{v}_\infty \cos \bar{\gamma}}{\dot{\sigma}} \quad (7.121)$$

one has the alternative **point-mass steady-flight equations**

$$\begin{bmatrix} \bar{T} \cos \bar{\alpha} - \bar{D} - mg \sin \bar{\gamma} \\ mg \sin \bar{\mu} \cos \bar{\gamma} \\ -\bar{T} \sin \bar{\alpha} - \bar{L} + mg \cos \bar{\mu} \cos \bar{\gamma} \end{bmatrix} = \begin{bmatrix} 0 \\ m \frac{(\bar{v}_\infty \cos \bar{\gamma})^2}{R_c} \cos \bar{\mu} \\ m \frac{(\bar{v}_\infty \cos \bar{\gamma})^2}{R_c} \sin \bar{\mu} \end{bmatrix} \quad (7.122)$$

It should be noted that from the second equation, i.e.

$$mg \sin \bar{\mu} \cos \bar{\gamma} = m \frac{(\bar{v}_\infty \cos \bar{\gamma})^2}{R} \cos \bar{\mu} \quad (7.123)$$

one can see that for any non-zero bank angle

$$\dot{\sigma} = \frac{g \tan \bar{\mu}}{\bar{v}_\infty} \quad (7.124)$$

and

$$R_c = \frac{\bar{v}_\infty^2 \cos \bar{\gamma}}{g |\tan \bar{\mu}|} \quad (7.125)$$

The most general maneuver described by the point-mass steady-flight equations is a steady climbing or descending coordinated turn and are primarily controlled by altering the lift, thrust, and bank angle of the airplane. The trajectory the airplane flies during this maneuver is a helix about the z_N -axis and a circular projection on the $x_N - y_N$ plane. Three special cases of this steady-flight maneuver are straight climbs/descents, level turns, and straight-and-level where **straight flight** occurs when $\dot{\sigma} = \bar{\mu} = 0^\circ$ and **level flight** occurs when $\bar{\gamma} = 0^\circ$. As the lift and drag are primarily a function of the steady-state air density, airspeed, and angle of attack, the point-mass steady-flight equations can be considered as a balance of six conditions: the altitude (affects air density), bank angle, flight-path angle, angle of attack, airspeed, and thrust for a given airplane's aerodynamic and mass properties. However, once one considers the airplane as a rigid body, the additional moment equations must also be balanced to ensure that the airplane remains at the prescribed steady-flight conditions. As these additional moment equations can be altered by the control inputs to the ailerons, rudder, and elevator, one typically refers to this moment balance as **trimming the airplane**, which will be considered in the subsequent chapter.

Point-Mass Spacecraft Steady-Flight

Orbital mechanics describe the classical mechanics derivations when modeling spacecraft as point-masses. This and other introductory sections of the textbook on spacecraft dynamics will assume that $M_{Earth} \gg m$ and the **two-body approximation**, i.e. that Earth is the only gravitational body affecting the spacecraft and thus, the ECI frame is an inertial reference frame. Then, in the absence of aerodynamic forces, the equation of motion of the spacecraft in **coast**, i.e. no propulsion, can be modeled by the force of gravity acting towards the center of the Earth, i.e.

$$\ddot{\vec{r}}_I = \frac{F_g}{m} \frac{\vec{r}_I}{\|\vec{r}_I\|_2} = -\frac{\mu}{\|\vec{r}_I\|_2^3} \vec{r}_I \quad (7.126)$$

where \vec{r}_I is the position vector of the spacecraft relative to the center of the Earth in ECI coordinates. This second-order equation of a three-dimensional vector can be solved explicitly given six initial conditions, e.g. $\vec{r}_0 = \vec{v}_0$ and $\dot{\vec{r}}_0$. Thus, if one knows the position and velocity of a spacecraft at any point, one can determine the reference trajectory of that spacecraft. Furthermore, one is typically interested in the **orbital angular momentum** defined as

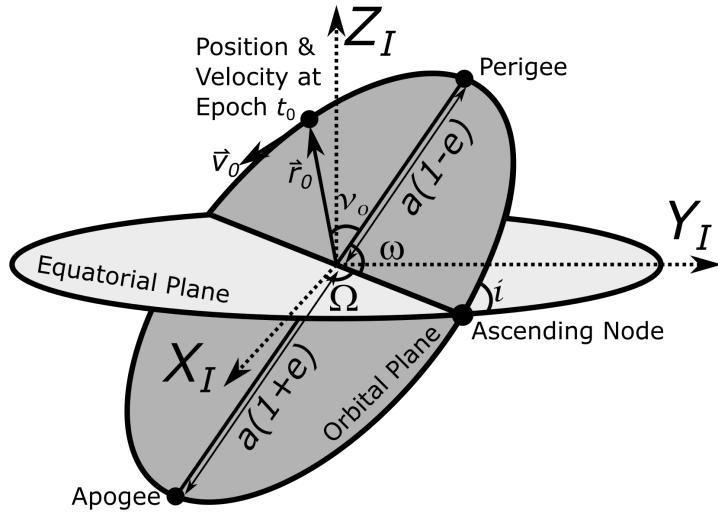
$$\vec{h}_I = [\vec{r}_I] \times \vec{v}_I \quad (7.127)$$

which by taking the cross product of the equation of motion with the angular momentum on both sides and integrating to obtain a constant of integration, one can show that the

$$\frac{[\vec{v}_I] \times \vec{h}_I}{\mu} - \frac{\vec{r}_I}{\|\vec{r}_I\|_2} = \vec{e} \quad (7.128)$$

where \vec{e} is called the **eccentricity vector** and whose magnitude, $e = \|\vec{e}\|_2$ determines the type of orbit for the spacecraft. Namely, if $0 \leq e < 1$, the spacecraft is in a *stable* elliptical orbit and if $e \geq 1$, the spacecraft is in a hyperbolic or parabolic orbit and will eventually escape the gravitational attraction of Earth, i.e. it is not a stable trajectory around Earth.

For spacecraft in an elliptical orbit, instead of specifying the position and velocity at some **epoch**, or reference time, t_0 , one can represent their reference trajectory about a celestial body by six **orbital elements**, also known as the **Keplerian elements**, which are depicted about that celestial body as



and defined as:

- the **eccentricity**, $0 < e < 1$, which is a measure of the elongation of the elliptical orbit compared to a circular orbit ($e = 0$);
- the **semi-major axis**, a , which is half the distance between the **periapsis** and **apoapsis**, i.e. the closest and further points from the focus corresponding to the planet's center of gravity;
- the **inclination**, i , which is the angle between the reference and orbital planes at the ascending node where the satellite moves from beneath the equatorial plane to above;
- the **longitude of ascending node**, Ω , which is the angle relative to the reference direction where the orbital plane intersects the reference plane;
- the **argument of periapsis**, ω , which is the angle in the orbital plane from the ascending node to periapsis; and
- the **true anomaly**, ν_0 at some **epoch**, or time, t_0 .

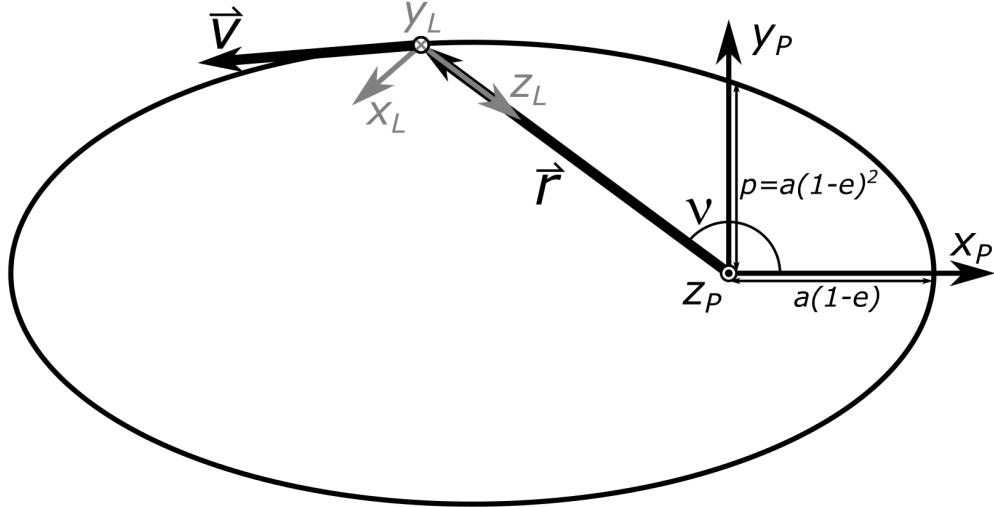
For **geocentric orbits**, i.e. the celestial body is Earth, ω is also known as the **argument of perigee** and Ω is the **right ascension of the ascending node** if the reference plane is the equatorial plane and the reference direction is the first point of Aries, i.e. the x -axis of the ECI frame.

The orbital plane can be described formally as the **perifocal frame** (subscript P), can be described as celestial body-centered

- the origin is at the focus corresponding to the celestial body's center of mass;

- the x_P -axis is colinear with a line connecting the two foci;
- the y_P -axis is orthogonal to x_P - and z_P -axes by the right hand rule; and
- the z_P -axis is normal to the orbital plane in the direction of the orbital angular momentum vector.

This frame can be depicted as



Notably, translation between the LVLH frame and the perifocal frame is \vec{r}_P and the direction cosine matrix for the LVLH frame of the satellite from the perifocal frame is given by

$$C_{L \leftarrow P} = C_1(90^\circ)C_2(-180^\circ)C_3(\nu) \quad (7.129)$$

Furthermore, the direction cosine matrix (DCM) for the perifocal frame, P , from the Earth-Centered Inertial reference frame, I , is given by

$$C_{P \leftarrow I} = C_3(\omega)C_1(i)C_3(\Omega) \quad (7.130)$$

where (ω, i, Ω) are the 3 – 1 – 3 Euler angles of the rotation.

Thus, if the perifocal frame is constant, i.e. $\dot{a} = \dot{e} = \dot{i} = \dot{\Omega} = \dot{\omega} = 0$, the position of the satellite in the perifocal frame is given by

$$\vec{r}_P = \begin{bmatrix} \frac{p \cos \nu}{1+e \cos \nu} \\ \frac{p \sin \nu}{1+e \cos \nu} \\ 0 \end{bmatrix} \quad (7.131)$$

where p is the **semi-latus rectum**

$$p = a(1 - e^2) \quad (7.132)$$

the velocity of the satellite in the perifocal frame is given by

$$\vec{v}_P = \begin{bmatrix} -\sqrt{\frac{\mu}{p}} \sin \nu \\ \sqrt{\frac{\mu}{p}}(e + \cos \nu) \\ 0 \end{bmatrix} \quad (7.133)$$

the angular velocity of the LVLH frame with respect to the perifocal frame can be expressed as

$$\vec{\omega}_{L/P} = C_1(90^\circ)C_2(-180^\circ)[\vec{r}_P] \times \vec{v}_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ \frac{n(1+e \cos \nu)^2}{(1-e^2)^{3/2}} \end{bmatrix} \quad (7.134)$$

or

$$\vec{\omega}_{L/P,L} = \begin{bmatrix} 0 \\ -\omega_O \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ -\frac{n(1+e \cos \nu)^2}{(1-e^2)^{3/2}} \\ 0 \end{bmatrix} \quad (7.135)$$

where ω_O is the **orbital angular velocity** and n is the **orbital mean motion** defined as

$$n = \frac{\mu}{a^3} \quad (7.136)$$

Lastly, the angular acceleration of the LVLH frame with respect to the perifocal frame can be expressed as

$$\vec{\alpha}_{L/P,L} = \begin{bmatrix} 0 \\ 0 \\ \alpha_O \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \frac{-2\mu e \sin \nu}{(\frac{p}{1+e \cos \nu})^3} \end{bmatrix} \quad (7.137)$$

where α_O is the **orbital angular acceleration**. Note that if one has a circular orbit of radius a , i.e., $e = 0$, then $\omega_O = n = \frac{\mu}{a^3}$ and $\alpha_O = 0$.

References

For more information, please refer to the following

- Curtis, H. D., “2.4 Angular Momentum and the Orbit Formulas,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 42-50
- Curtis, H. D., “2.7 Elliptical Orbits ($0 < e < 1$)”, *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 76-80
- Curtis, H. D., “4.6 Transformation between Geocentric Equatorial and Perifocal Frames,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 172-176
- Pesce, V., Colagrossi, A., and Silvestrini, S., “Chapter Two - Reference Systems and Planetary Models,” *Modern Spacecraft Guidance, Navigation, and Control*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2023, pp. 45-75
- Saarlas, M., “Appendix B: On the Drag Coefficient,” *Aircraft Performance*, 1st ed., Vol. 1, John Wiley & Sons, 2007, pp. 260-264
- Schmidt, D. K., “2.7 Point-Mass Performance Equations,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 76-80

Introductory Rigid Aerospace Vehicle Dynamics

8.1 Introductory Rigid Aerospace Vehicle Dynamics

Rigid-body dynamics are represented using the **Newton-Euler equations of motion (EOM)** which adds Euler's equation governing rotation which relates moments acting on the rigid body to the angular momentum. This results in the vector-valued differential equations:

$$\begin{aligned}\sum \vec{F} &= \frac{d}{dt} \vec{p}_G \\ \sum \vec{M} &= \frac{d}{dt} \vec{H}_G\end{aligned}\tag{8.1}$$

where $\sum \vec{M}$ is the total moment on the rigid body, and \vec{H}_G is the angular momentum about the center of mass G . It should be noted that the first equation of the Newton-Euler EOMs is called the **translation equation** and is equivalent to the point-mass equation while the second is called the **rotation equation**.

Since a rigid body can be represented as a continuous mass distribution, these quantities can be computed as integrals over the volume V by the following equations

$$\begin{aligned}\int_V \vec{f} \rho dV &= \frac{d}{dt} \int_V \vec{v} \rho dV \\ \int_V \vec{r} \times \vec{f} \rho dV &= \frac{d}{dt} \int_V \vec{r} \times \vec{v} \rho dV\end{aligned}\tag{8.2}$$

where \vec{f} is the forces acting on the rigid body per unit mass, $dm = \rho dV$ is an infinitesimal mass element of the body with density ρ , \vec{v} is the velocity of that mass element, and \vec{r} is the position vector of the mass element with respect to the origin of an inertial reference frame.

To get around these integrals, one can represent the static distribution of mass through the **inertia matrix**, also known as the **inertia tensor**, which is composed of the **moments of inertia** and the **products of inertia**

about the body-fixed frame coordinates axes $x_B - y_B - z_B$, i.e.

$$I_G = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix}, \quad (8.3)$$

where

$$I_{xx} = \int_V (y_B^2 + z_B^2) \rho dV \quad (8.4)$$

$$I_{yy} = \int_V (x_B^2 + z_B^2) \rho dV \quad (8.5)$$

$$I_{zz} = \int_V (x_B^2 + y_B^2) \rho dV \quad (8.6)$$

$$I_{xy} = I_{yx} = \int_V x_B y_B \rho dV \quad (8.7)$$

$$I_{xz} = I_{zx} = \int_V x_B z_B \rho dV \quad (8.8)$$

$$I_{yz} = I_{zy} = \int_V y_B z_B \rho dV \quad (8.9)$$

This introductory chapter on rigid-body dynamics will also use the **constant-mass approximation** which assumes the mass distribution of the rigid aerospace vehicle does not change. However, for non-electric powered aerospace vehicles, engine fuel consumption will cause mass to change at some rate. This mass rate may be slow in the case of aircraft engines and is often neglected, but for rocket engines the mass rate is crucial in deriving appropriate EOMs. This and other mass effects will be addressed in the next chapter of this textbook on advanced rigid aerospace vehicle dynamics.

Thus, if one assumes that the total mass m and inertia matrix are constant over time and one defines the reference frame as a body-fixed frame, i.e. attached to the rigid structure, then the Newton-Euler EOMs written in the rotating reference frame become

$$\begin{aligned} \sum \vec{F}_B &= m (\dot{\vec{v}}_B + \vec{\omega}_{B/I,B} \times \vec{v}_B) \\ \sum \vec{M}_B &= I_G \vec{\omega}_{B/I,B} + \vec{\omega}_{B/I,B} \times I_G \vec{\omega}_{B/I,B} \end{aligned} \quad (8.10)$$

where $\sum \vec{F}_B$ is the total force acting on the center of mass G in the body-fixed frame, \vec{v}_B is the linear velocity of the center of mass, $\sum \vec{M}_B$ is the total moment acting about the center of mass G in the body-fixed frame, and $\vec{\omega}_{B/I,B}$ is the angular velocity of the body-fixed frame relative to the inertial frame written in body-frame coordinates.

For the body-fixed frame linear and angular velocity components, one typically uses the following representations

$$\vec{v}_B = [u \ v \ w]^T \quad (8.11)$$

and

$$\vec{\omega}_{B/I,B} = [p \ q \ r]^T \quad (8.12)$$

which allows one to write out the translation equation using a skew-symmetric matrix as

$$\sum \vec{F}_B = m \left(\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \right) \quad (8.13)$$

$$\sum \vec{F}_B = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (8.14)$$

and the rotation equation using a skew-symmetric matrix as

$$\sum \vec{M}_B = \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & -I_{xy} & -I_{xz} \\ -I_{yx} & I_{yy} & -I_{yz} \\ -I_{zx} & -I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (8.15)$$

$$\sum \vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} - I_{xy}\dot{q} - I_{xz}\dot{r} \\ -I_{xy}\dot{p} + I_{yy}\dot{q} - I_{yz}\dot{r} \\ -I_{xz}\dot{p} - I_{yz}\dot{q} + I_{zz}\dot{r} \end{bmatrix} + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx}p - I_{xy}q - I_{xz}r \\ -I_{xy}p + I_{yy}q - I_{yz}r \\ -I_{xz}p - I_{yz}q + I_{zz}r \end{bmatrix} \quad (8.16)$$

$$\sum \vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xy}(\dot{q} - pr) - I_{xz}(\dot{r} + pq) - I_{yz}(q^2 - r^2) \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - I_{xy}(\dot{p} + qr) - I_{xz}(r^2 - p^2) - I_{yz}(\dot{r} - pq) \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq - I_{xy}(p^2 - q^2) - I_{xz}(\dot{p} - qr) - I_{yz}(\dot{q} + pr) \end{bmatrix} \quad (8.17)$$

Thus, for any rigid body in a rotating body-fixed frame, the Newton-Euler EOMs are

$$\sum \vec{F}_B = m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix}$$

$$\sum \vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xy}(\dot{q} - pr) - I_{xz}(\dot{r} + pq) + I_{yz}(r^2 - q^2) \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - I_{xy}(\dot{p} + qr) + I_{xz}(p^2 - r^2) - I_{yz}(\dot{r} - pq) \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + I_{xy}(q^2 - p^2) - I_{xz}(\dot{p} - qr) - I_{yz}(\dot{q} + pr) \end{bmatrix} \quad (8.18)$$

These represent six coupled nonlinear ODEs with 6 free variables, u , v , w , p , q , r , thus forming the **six degree-of-freedom (6-DOF) equations of motion (EOM)**.

It should be noted that the directions of the forces and moments for aerospace vehicles are typically expressed well in different frames than the body-fixed frame. Thus, these terms will generally depend on the relative passive rotations and the representation used, e.g., the DCM, the Euler symmetric parameters, or the Euler angles, all of which change as a function of the angular velocity resulting in supplementary differential equations relating the angular velocity to the passive rotations. This textbook will use either the 3 – 1 – 3 or 3 – 2 – 1 Euler angles to represent these dynamics.

Thus, if the Euler angles are used to represent the attitude relative to the inertial frame, one has for the 3 – 1 – 3 Euler angles

$$\begin{bmatrix} \omega_p \\ \omega_n \\ \omega_s \end{bmatrix} = \begin{bmatrix} \dot{\phi}_p \\ \dot{\theta}_n \\ \dot{\psi}_s \end{bmatrix} = \begin{bmatrix} \sin \psi_s \csc \theta_n & \cos \psi_s \csc \theta_n & 0 \\ \cos \psi_s & -\sin \psi_s & 0 \\ -\sin \psi_s \cot \theta_n & -\cos \psi_s \cot \theta_n & 1 \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (8.19)$$

and for the 3 – 2 – 1 Euler angles

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (8.20)$$

However, this equation will change if the navigation frame is not considered inertial which is developed in the subsequent chapter of this textbook.

Rigid Airplane Equations of Motion

By convention, the aerodynamic and propulsive forces along the x_B -, y_B -, and z_B -axes are combined and normalized by the mass of the aircraft, m , and denoted by X , Y , and Z , respectively. This infers the following force equation for aircraft as

$$\vec{F}_{p,B} + \vec{F}_{a,B} = \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} \quad (8.21)$$

or

$$\frac{1}{m} (\vec{F}_{p,B} + \vec{F}_{a,B}) = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (8.22)$$

As an example, one could model this for airplanes as

$$\frac{1}{m} \left(\begin{bmatrix} T \cos \theta_T \\ 0 \\ T \sin \theta_T \end{bmatrix} + \begin{bmatrix} -D \cos \alpha \cos \beta - S \cos \alpha \sin \beta + L \sin \alpha \\ S \cos \beta - D \sin \beta \\ -D \sin \alpha \cos \beta - S \sin \alpha \sin \beta - L \cos \alpha \end{bmatrix} \right) = \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} \quad (8.23)$$

Furthermore, the aerodynamic and propulsive moments about the x_B -, y_B -, and z_B -axes are conventionally normalized by the moments of inertia, I_{xx} , I_{yy} , and I_{zz} , about that axis, i.e. x_B , y_B , and z_B , and denoted by L , M , and N , respectively. This infers the following moment equation for aircraft as

$$\vec{M}_{p,B} + \vec{M}_{a,B} = \begin{bmatrix} I_{xx} L_{roll} \\ I_{yy} M \\ I_{zz} N \end{bmatrix} \quad (8.24)$$

$$\vec{M}_{p,B} + \vec{M}_{a,B} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} L_{roll} \\ M \\ N \end{bmatrix} \quad (8.25)$$

As an example, one could model this for airplanes as

$$\begin{bmatrix} I_{xx}^{-1} & 0 & 0 \\ 0 & I_{yy}^{-1} & 0 \\ 0 & 0 & I_{zz}^{-1} \end{bmatrix} \left(\begin{bmatrix} 0 \\ T(z_T \cos \theta_T - x_T \sin \theta_T) \\ 0 \end{bmatrix} + \vec{M}_{a,B} \right) = \begin{bmatrix} L_{roll} \\ M \\ N \end{bmatrix} \quad (8.26)$$

Furthermore, the vast majority of aircraft are designed as symmetric in the $x_B - z_B$ plane, i.e. the left side of the aircraft is reflected to the right side, thus the inertia matrix can be simplified to

$$I_G = \begin{bmatrix} I_{xx} & 0 & -I_{xz} \\ 0 & I_{yy} & 0 \\ -I_{xz} & 0 & I_{zz} \end{bmatrix} \quad (8.27)$$

where often I_{xz} is also neglected due to its relatively small magnitude.

For aircraft, the convention is to assume the 3–2–1 Euler angle representation for relating the navigation frame to the body-fixed frame for modeling the gravitational force. Assuming a flat- or spherical-Earth approximation, one can mode the gravity as

$$\vec{F}_{g,B} = C_{B \leftarrow N} \vec{F}_{g,N} = \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (8.28)$$

$$C_{B \leftarrow N} \vec{F}_{g,N} = mg \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} \quad (8.29)$$

which results in

$$\begin{aligned} \vec{F}_{a,B} + \vec{F}_{p,B} + mg \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} &= m \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \\ \vec{M}_{a,B} + \vec{M}_{p,B} &= \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr - I_{xy}(\dot{q} - pr) - I_{xz}(\dot{r} + pq) + I_{yz}(r^2 - q^2) \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr - I_{xy}(\dot{p} + qr) + I_{xz}(p^2 - r^2) - I_{yz}(\dot{r} - pq) \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq + I_{xy}(q^2 - p^2) - I_{xz}(\dot{p} - qr) - I_{yz}(\dot{q} + pr) \end{bmatrix} \end{aligned} \quad (8.30)$$

which notably only requires that eight states be known to calculate the derivatives at any instant, namely u , v , w , p , q , r , ϕ , and θ , while ψ is simply a derived parameter from p , q , and r .

Finally, with these substitutions and zeroing the solar forces and moments, one obtains the **rigid aircraft equations of motion**

$$\begin{aligned} \begin{bmatrix} X - g \sin \theta \\ Y + g \sin \phi \cos \theta \\ Z - g \cos \phi \cos \theta \end{bmatrix} &= \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \\ \begin{bmatrix} L \\ M \\ N \end{bmatrix} &= \begin{bmatrix} \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} pr + \frac{I_{xz}}{I_{yy}} (p^2 - r^2) \\ \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \end{aligned} \quad (8.31)$$

This introductory chapter of the textbook will use the **no-wind approximation** to introduce rigid aircraft dynamics. Although with wind speeds significantly close to the nominal aircraft airspeed and/or long distance flight analysis, the no-wind approximation may not be suitable. The additional effects of both steady and unsteady wind on the aircraft dynamics is presented in the next chapter of this textbook on advanced rigid

aerospace vehicle dynamics. As an alternative to this form of the EOMs with the no-wind approximation, then $\vec{v}_g = \vec{v}_\infty$, and one may also substitute for the y_B -axis and z_B -axis velocity terms

$$v = u \tan \beta \quad (8.32)$$

$$w = u \sin \alpha \quad (8.33)$$

as well as their derivatives

$$\dot{v} = \dot{u} \tan \beta + \dot{\beta} u \sec^2 \beta \quad (8.34)$$

$$\dot{w} = \dot{u} \sin \alpha + \dot{\alpha} u \cos \alpha \quad (8.35)$$

in order to obtain the **alternative rigid aircraft EOMs**

$$\begin{aligned} \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \vec{F}_G &= \begin{bmatrix} \dot{u} + qu \sin \alpha - ru \tan \beta \\ \dot{u} \tan \beta + \dot{\beta} u \sec^2 \beta + ru - pu \sin \alpha \\ \dot{u} \sin \alpha + \dot{\alpha} u \cos \alpha + pu \tan \beta - qu \end{bmatrix} \\ \begin{bmatrix} L \\ M \\ N \end{bmatrix} &= \begin{bmatrix} \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \end{aligned} \quad (8.36)$$

Lastly, one is typically also interested in the position of the rigid body in the navigation frame. Letting $[\dot{x}_N \dot{y}_N \dot{z}_N]^T$ represent the velocity in the navigation frame, these quantities can be related to the previous body-fixed frame by the simple rotation

$$\begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} = C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (8.37)$$

or

$$\begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (8.38)$$

where notably one can also use $\dot{z}_N = -\dot{h}$. Using this representation in the navigation frame, it is then possible to integrate the linear velocity to obtain the trajectory of the aircraft for a flat-Earth model, $\vec{x}_N(t)$. The integration of the velocity typically is done numerically because ϕ , θ , and ψ are also functions of time.

Rigid Spacecraft Attitude Equations of Motion

As spacecraft are typically designed in a wide variety of shapes, the inertia tensor varies depending on where one places the body-fixed frame axes. The **principal body-fixed frame** is defined as the axes for which the inertia tensor is diagonal, i.e.

$$I_G = \begin{bmatrix} I_1 & 0 & 0 \\ 0 & I_2 & 0 \\ 0 & 0 & I_3 \end{bmatrix} \quad (8.39)$$

where I_1 , I_2 , and I_3 are the **principal moments of inertia**. Where the largest to smallest in magnitude are known as the major, intermediate, and minor moments of inertia and axes. Note that given another body-fixed frame with inertia tensor, I'_G , one can use the eigenvalue decomposition of the *symmetric* matrix I'_G to form I_G by

$$I_G = V I'_G V^T \quad (8.40)$$

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} = [\vec{v}_1 \quad \vec{v}_2 \quad \vec{v}_3] I'_G [\vec{v}_1^T \quad \vec{v}_2^T \quad \vec{v}_3^T] \quad (8.41)$$

where I_{xx} , I_{yy} , and I_{zz} are the eigenvalues of I'_G with corresponding right eigenvectors, \vec{v}_1 , \vec{v}_2 , \vec{v}_3 , respectively.

Thus, for the angular momentum of the satellite in principal body-fixed frame coordinates, one has

$$\vec{H}_{G,B} = I_G \vec{\omega}_{B/I,B} = \begin{bmatrix} I_{xx}p \\ I_{yy}q \\ I_{zz}r \end{bmatrix} \quad (8.42)$$

Finally, assuming principal body-fixed frame for the Newton-Euler equations of motion, one obtains the **satellite attitude equations of motion**

$$\vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq \end{bmatrix} \quad (8.43)$$

Without loss of generality, one can assume \vec{H}_G defines the z_I -axis as the orbital plane is assumed to be static. Also, note that the translation of the satellite is only governed by orbital mechanics as described in the inertial frame and is decoupled from the attitude dynamics. Here the inertial velocity can be rotated to body-fixed frame if one wishes to express the apparent body-fixed frame velocities, typically using the orbital elements as described previously and the angular velocities from the attitude dynamics.

References

For more information, please refer to the following

- Curtis, H. D., “9.3 Equations of Translational Motion,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 408-410
- Curtis, H. D., “9.4 Equations of Rotational Motion,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 420-414
- Curtis, H. D., “9.5 Moments of Inertia,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 414-435
- Nelson, R. C., “3.2 Derivation of Rigid Body Equations of Motion,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 49-434

- Nelson, R. C., “3.3 Orientation and Position of the Airplane,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 101-103
- Schmidt, D. K., “2.3 Reference and Perturbation Equations - Flat Earth,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 42-48
- Sidi, M. J., “4.2 Angular Momentum and the Inertia Matrix,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 88-90
- Sidi, M. J., “4.4 Moment-of-Inertia Matrix in Selected Axis Frames,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 90-95
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “1.7 Rigid-Body Dynamics,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 34-43

8.2 Introductory Rigid Airplane Trim Analysis

Rigid-Body Steady-Flight Equations

When modeling an airplane as a rigid-body, the **rigid-body equilibrium flight conditions**, also known as the **rigid-body steady-flight conditions**, by definition, occur when the state variables in the rigid airplane EOMs are constant, i.e.

$$\dot{u} = \dot{\alpha} = \dot{\beta} = \dot{p} = \dot{q} = \dot{r} = \dot{\phi} = \dot{\theta} = 0 \quad (8.44)$$

which imply that the steady-flight conditions solve the **rigid-body steady-flight equations** in the body-fixed frame as

$$\begin{bmatrix} \bar{X} - g \sin \bar{\theta} \\ \bar{Y} + g \sin \bar{\phi} \cos \bar{\theta} \\ \bar{Z} - g \cos \bar{\phi} \cos \bar{\theta} \\ \bar{L}_{roll} \\ \bar{M} \\ \bar{N} \end{bmatrix} = \begin{bmatrix} \bar{q}\bar{u} \sin \bar{\alpha} - \bar{r}\bar{u} \tan \bar{\beta} \\ \bar{r}\bar{u} - \bar{p}\bar{u} \sin \bar{\alpha} \\ \bar{p}\bar{u} \tan \bar{\beta} - \bar{q}\bar{u} \\ \frac{I_{zz}-I_{yy}}{I_{xx}} \bar{q}\bar{r} - \frac{I_{xz}}{I_{xx}} \bar{p}\bar{q} \\ \frac{I_{xx}-I_{zz}}{I_{yy}} \bar{p}\bar{r} + \frac{I_{xz}}{I_{yy}} (\bar{p}^2 - \bar{r}^2) \\ \frac{I_{yy}-I_{xx}}{I_{zz}} \bar{p}\bar{q} + \frac{I_{xz}}{I_{zz}} \bar{q}\bar{r} \end{bmatrix} \quad (8.45)$$

and the supplemental kinematic equations given as

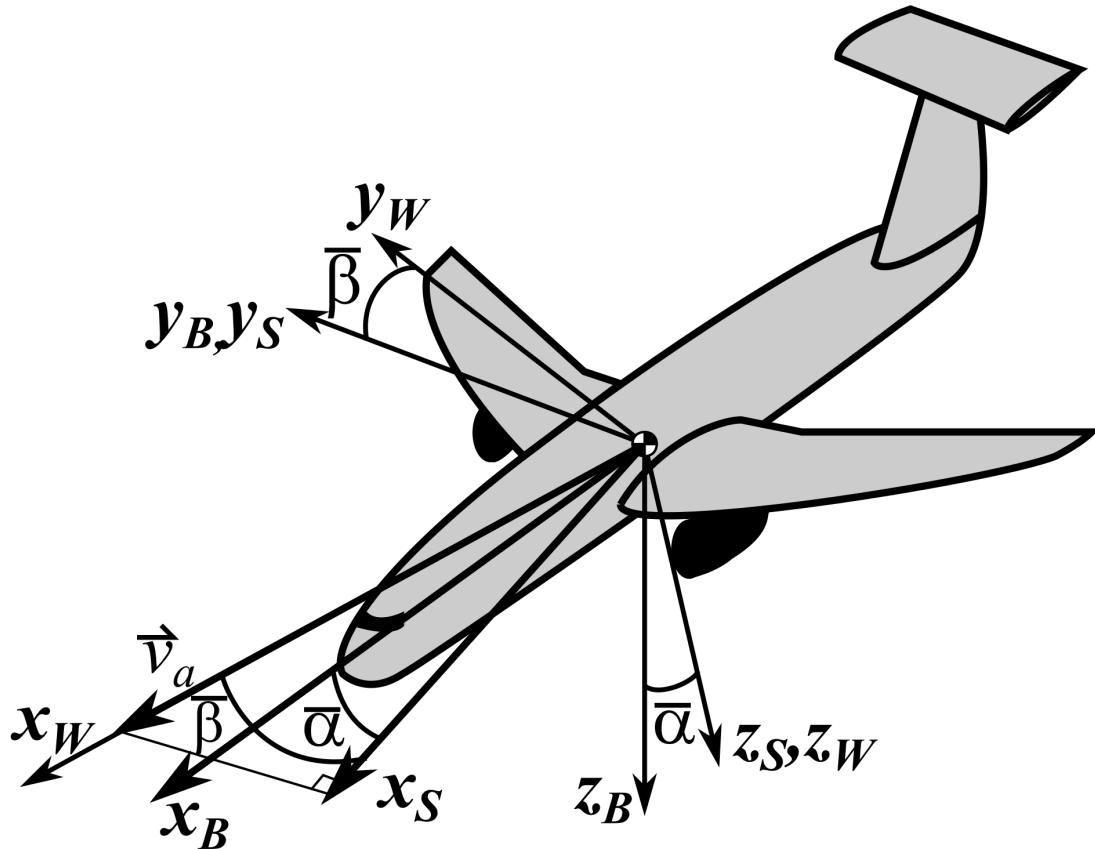
$$\begin{bmatrix} 0 \\ 0 \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} \bar{p} + \bar{q} \sin \bar{\phi} \tan \bar{\theta} + \bar{r} \cos \bar{\phi} \tan \bar{\theta} \\ \bar{q} \cos \bar{\phi} - \bar{r} \sin \bar{\phi} \\ \bar{q} \sin \bar{\phi} \sec \bar{\theta} + \bar{r} \cos \bar{\phi} \sec \bar{\theta} \end{bmatrix} \quad (8.46)$$

or

$$\begin{bmatrix} \bar{p} \\ \bar{q} \\ \bar{r} \end{bmatrix} = \begin{bmatrix} -\dot{\psi} \sin \bar{\theta} \\ \dot{\psi} \sin \bar{\phi} \cos \bar{\theta} \\ \dot{\psi} \cos \bar{\phi} \cos \bar{\theta} \end{bmatrix} \quad (8.47)$$

where the \bar{u} , $\bar{\beta}$, $\bar{\alpha}$, \bar{p} , \bar{q} , \bar{r} , $\bar{\phi}$, $\bar{\theta}$, and $\dot{\psi}$ are the steady-flight conditions while \bar{X} , \bar{Y} , \bar{Z} , \bar{L} , \bar{M} , and \bar{N} are the aerodynamic and propulsive forces and moments at steady-flight and are functions of the steady-flight conditions. Alternatively, one may use the thrust, lift, side, and drag forces at steady flight, \bar{T} , \bar{L} , \bar{S} , and \bar{D} , respectively, instead of \bar{X} , \bar{Y} , and \bar{Z} . It should also be noted that the gravitational acceleration, g , and air density, ρ , will also vary as a function of altitude, h , which further requires that a strictly steady-flight condition would be at a constant altitude. However, as these variations occur slowly, one typically assumes these are constant for analyzing different “steady-flight” maneuvers.

In addition, recall that previous modeling defined the aerodynamic and propulsive forces in a body-fixed frame affixed to fuselage, also known as a **fuselage frame**. However, for the linearized rigid airplane dynamics, it is common to use an alternative body-fixed frame known as the **stability frame** (subscript S). In particular, the stability frame is related to the fuselage frame through a rotation of $\bar{\alpha}$ about the y_B -axis as shown in the following figure.



Note that by being defined by $\bar{\alpha}$, *different* stability frames are defined for *different* steady-flight conditions. However, the stability frame rotates with the airplane body as the perturbed states vary, not remaining fixed to the free-stream velocity vector as for the wind frame. Furthermore, it should be noted that $\theta = \gamma$ if θ describes the pitch angle from N to S .

It is important to note that here the normalized force and moment vector elements, X , Y , Z , L_{roll} , M , and N , are now defined in the stability frame, but can be related to the fuselage-fixed body frame B using a C_2 basic rotation matrix about $\bar{\alpha}$, i.e.

$$\vec{v}_B = C_2(\bar{\alpha}) \vec{v}_S \quad \vec{v}_S = C_2^T(\bar{\alpha}) \vec{v}_B \quad (8.48)$$

where \vec{v} is some vector. In addition, the **inertia matrix in the stability frame**, I_S , is related to the inertia matrix in fuselage-fixed frame, I_B , through the transformation

$$I_B = C_2(\bar{\alpha}) I_S C_2^T(\bar{\alpha}) \quad I_S = C_2^T(\bar{\alpha}) I_B C_2(\bar{\alpha}) \quad (8.49)$$

Straight Steady-Flight Equations

For straight steady-flight, i.e., $\dot{\psi} = 0^\circ/\text{s}$, represented in the stability frame, i.e., $\bar{\alpha} = 0^\circ$, one has

$$\begin{bmatrix} \bar{p} \\ \bar{q} \\ \bar{r} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (8.50)$$

Thus, with the substitution $\bar{\theta} = \bar{\gamma}$ which holds in the stability frame, one has

$$\begin{bmatrix} \bar{X} - g \sin \bar{\gamma} \\ \bar{Y} + g \sin \bar{\phi} \cos \bar{\gamma} \\ \bar{Z} - g \cos \bar{\phi} \cos \bar{\gamma} \\ \bar{L}_{roll} \\ \bar{M} \\ \bar{N} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8.51)$$

Next, substituting for the propulsive and aerodynamic forces separately in the stability frame, one has

$$\begin{bmatrix} -\bar{D} \cos \bar{\beta} - \bar{S} \sin \bar{\beta} + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ \bar{S} \cos \bar{\beta} - \bar{D} \sin \bar{\beta} \\ \bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) \\ \bar{L}_{a,S} \\ \bar{M}_{a,S} + \bar{T}(z_T \cos(\theta_T + \bar{\alpha}) - x_T \sin(\theta_T + \bar{\alpha})) \\ \bar{N}_{a,S} + \bar{N}_{p,S} \end{bmatrix} = \begin{bmatrix} mg \sin \bar{\gamma} \\ -mg \sin \bar{\phi} \cos \bar{\gamma} \\ mg \cos \bar{\phi} \cos \bar{\gamma} \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad (8.52)$$

where $\bar{N}_{p,S} \neq 0$ has been included in the case of an **engine-out flight condition** which occurs for multiple-engine airplanes. These are typically separated into the **longitudinal straight steady-flight equations**

$$\begin{bmatrix} -\bar{D} \cos \bar{\beta} - \bar{S} \sin \bar{\beta} + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ \bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) \\ \bar{M}_{a,S} + \bar{T}(z_T \cos(\theta_T + \bar{\alpha}) - x_T \sin(\theta_T + \bar{\alpha})) \end{bmatrix} = \begin{bmatrix} mg \sin \bar{\gamma} \\ mg \cos \bar{\phi} \cos \bar{\gamma} \\ 0 \end{bmatrix} \quad (8.53)$$

and the **lateral-directional straight steady-flight equations**

$$\begin{bmatrix} \bar{S} \cos \bar{\beta} - \bar{D} \sin \bar{\beta} \\ \bar{L}_{a,S} \\ \bar{N}_{a,S} + \bar{N}_{p,S} \end{bmatrix} = \begin{bmatrix} -mg \sin \bar{\phi} \cos \bar{\gamma} \\ 0 \\ 0 \end{bmatrix} \quad (8.54)$$

Straight Longitudinal Trim Analysis

For simplified analysis, assume **wings-level**, i.e. $\bar{\phi} = 0^\circ$ and **coordinated flight**, i.e. $\bar{\beta} = 0^\circ$ and $\bar{N}_{p,S} = 0$, then one has only three non-trivial equations

$$\begin{bmatrix} -\bar{D} + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ \bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) \\ \bar{M}_a + \bar{T}(z_T \cos(\theta_T + \bar{\alpha}) - x_T \sin(\theta_T + \bar{\alpha})) \end{bmatrix} = \begin{bmatrix} mg \sin \bar{\gamma} \\ mg \cos \bar{\gamma} \\ 0 \end{bmatrix} \quad (8.55)$$

which describe the longitudinal trim of the airplane. Next, one can express the aerodynamic forces in terms of the coefficients, C_L , C_D , and C_m , as

$$\begin{bmatrix} -Q_\infty S_w \bar{C}_D + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ Q_\infty S_w \bar{C}_L + \bar{T} \sin(\theta_T + \bar{\alpha}) \\ Q_\infty S_w \bar{c}_w \bar{C}_m + \bar{T}(z_T \cos(\theta_T + \bar{\alpha}) - x_T \sin(\theta_T + \bar{\alpha})) \end{bmatrix} = \begin{bmatrix} mg \sin \bar{\gamma} \\ mg \cos \bar{\gamma} \\ 0 \end{bmatrix} \quad (8.56)$$

Furthermore, for a simplified analysis of these equations, one can assume linear relationships for these aerodynamic coefficients with respect to the trim angle of attack, $\bar{\alpha}$, and trim elevator deflection, $\bar{\delta}_e$, i.e.

$$\bar{C}_D = C_{D_0} + C_{D_\alpha} \bar{\alpha} + C_{D_{\delta_e}} \bar{\delta}_e \quad (8.57)$$

$$\bar{C}_L = C_{L_0} + C_{L_\alpha} \bar{\alpha} + C_{L_{\delta_e}} \bar{\delta}_e \quad (8.58)$$

and

$$\bar{C}_m = C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_{\delta_e}} \bar{\delta}_e \quad (8.59)$$

Then, by substitution, one has

$$\begin{bmatrix} -Q_\infty S_w (C_{D_0} + C_{D_\alpha} \bar{\alpha} + C_{D_{\delta_e}} \bar{\delta}_e) + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ Q_\infty S_w (C_{L_0} + C_{L_\alpha} \bar{\alpha} + C_{L_{\delta_e}} \bar{\delta}_e) + \bar{T} \sin(\theta_T + \bar{\alpha}) \\ Q_\infty S_w \bar{c}_w (C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_{\delta_e}} \bar{\delta}_e) + \bar{T} \frac{z_T \cos(\theta_T + \bar{\alpha}) - x_T \sin(\theta_T + \bar{\alpha})}{Q_\infty S_w \bar{c}_w} \end{bmatrix} = \begin{bmatrix} mg \sin \bar{\gamma} \\ mg \cos \bar{\gamma} \\ 0 \end{bmatrix} \quad (8.60)$$

Thus, for a given Q_∞ , i.e. a given airspeed and altitude, and flight-path angle, $\bar{\gamma}$, these equations determine the three unknowns for trimming the airplane, i.e. $\bar{\alpha}$, $\bar{\delta}_e$, and \bar{T} . This can be solved using numerical methods. Two analytical methods to determine approximate trim values is to assume $\bar{L} \gg \bar{T} \sin(\theta_T + \bar{\alpha})$ or $\sin(\theta_T + \bar{\alpha}) \approx 0$ as well as $\cos(\theta_T + \bar{\alpha}) \approx 1$, then one has

$$\begin{bmatrix} -C_{D_\alpha} & -C_{D_{\delta_e}} & \frac{1}{Q_\infty S_w} \\ C_{L_\alpha} & C_{L_{\delta_e}} & 0 \\ C_{m_\alpha} & C_{m_{\delta_e}} & \frac{z_T}{Q_\infty S_w \bar{c}_w} \end{bmatrix} \begin{bmatrix} \bar{\alpha} \\ \bar{\delta}_e \\ \bar{T} \end{bmatrix} \approx \begin{bmatrix} \frac{mg}{Q_\infty S_w} \sin \bar{\gamma} + C_{D_0} \\ \frac{mg}{Q_\infty S_w} \cos \bar{\gamma} - C_{L_0} \\ -C_{m_0} \end{bmatrix} \quad (8.61)$$

which can be solved by multiplying both sides by the inverse matrix on the left side. However, if one alternatively assumes $z_T \cos \theta_T - x_T \sin \theta_T \approx 0$, then one has a second method via

$$\begin{bmatrix} -C_{D_\alpha} & -C_{D_{\delta_e}} & \frac{\cos(\theta_T + \bar{\alpha})}{Q_\infty S_w} \\ C_{L_\alpha} & C_{L_{\delta_e}} & 0 \\ C_{m_\alpha} & C_{m_{\delta_e}} & 0 \end{bmatrix} \begin{bmatrix} \bar{\alpha} \\ \bar{\delta}_e \\ \bar{T} \end{bmatrix} \approx \begin{bmatrix} \frac{mg}{Q_\infty S_w} \sin \bar{\gamma} + C_{D_0} \\ \frac{mg}{Q_\infty S_w} \cos \bar{\gamma} - C_{L_0} \\ -C_{m_0} \end{bmatrix} \quad (8.62)$$

which allows one to decouple $\bar{\alpha}$ and $\bar{\delta}_e$ from \bar{T} . This results in the analytical solution

$$\begin{aligned}\bar{\alpha} &\approx \frac{\left(\frac{mg}{Q_\infty S_w} \cos \bar{\gamma} - C_{L_0}\right) C_{m_{\delta_e}} + C_{m_0} C_{L_{\delta_e}}}{C_{L_\alpha} C_{m_{\delta_e}} - C_{m_\alpha} C_{L_{\delta_e}}} \\ \bar{\delta}_e &\approx -\frac{\left(\frac{mg}{Q_\infty S_w} \cos \bar{\gamma} - C_{L_0}\right) C_{m_\alpha} + C_{m_0} C_{L_\alpha}}{C_{L_\alpha} C_{m_{\delta_e}} - C_{m_\alpha} C_{L_{\delta_e}}}\end{aligned}\quad (8.63)$$

which is only a function of the lift and M -moment coefficients. Then, one can determine the trim thrust via the original equation

$$\bar{T} \approx \frac{mg \sin \bar{\gamma} + (C_{D_0} + C_{D_\alpha} \bar{\alpha} + C_{D_{\delta_e}} \bar{\delta}_e) Q_\infty S_w}{\cos(\theta_T + \bar{\alpha})} \quad (8.64)$$

which notably is a function of the drag coefficients and the trim angle of attack and elevator deflection.

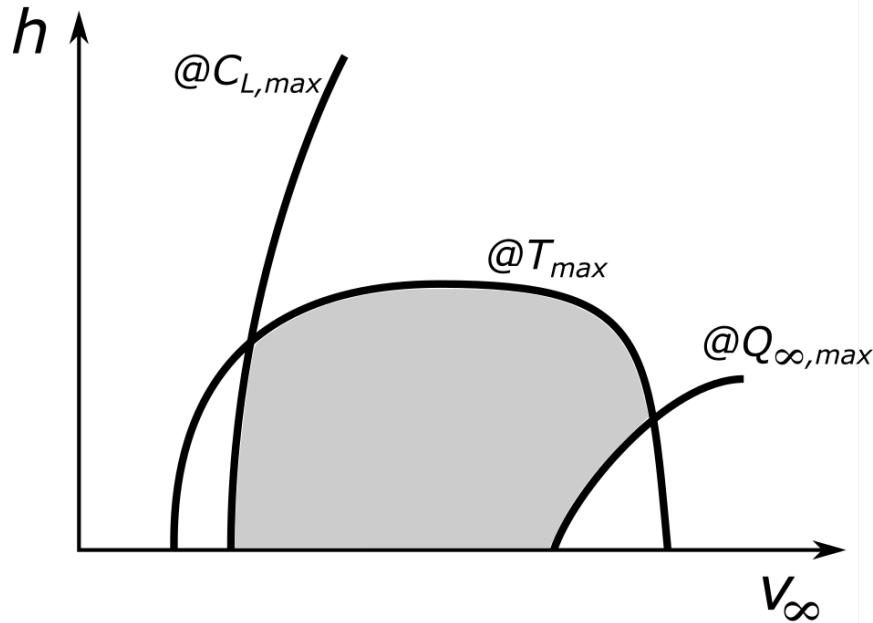
Lastly, recall that

$$\bar{L} = Q_\infty S_w \bar{C}_L \quad (8.65)$$

or

$$\bar{L} = Q_\infty S_w (C_{L_0} + C_{L_\alpha} \bar{\alpha} + C_{L_{\delta_e}} \bar{\delta}_e) \quad (8.66)$$

Thus, as these three trim values are coupled, the maximum limit on thrust, T_{max} , is the limiting factor of achievable lift via maximizing dynamic pressure, $Q_{\infty,max}$, and the maximum limit on elevator deflection, $\delta_{e,max}$ or angle of attack α_{max} is the limiting factor of achievable lift via maximizing the lift coefficient, $C_{L,max}$. These limits for straight-and-level flight, i.e. $\bar{\gamma} = 0$, define the vehicle's **flight envelope plot** which depicts the three curves of T_{max} , $C_{L,max}$, and $Q_{\infty,max}$ for airspeed, v_∞ versus altitude, h .



where the center region defines the possible steady-flight conditions for the airplane.

Straight Lateral-Directional Trim Analysis

One can express the lateral-directional aerodynamic forces and moments in terms of the coefficients, C_S , C_l , and C_n , as

$$\begin{bmatrix} Q_\infty S_w \bar{C}_S \cos \bar{\beta} - Q_\infty S_w \bar{C}_D \sin \bar{\beta} \\ Q_\infty S_w \bar{C}_l \\ Q_\infty S_w \bar{C}_n + \bar{N}_{p,S} \end{bmatrix} = \begin{bmatrix} -mg \sin \bar{\phi} \cos \bar{\gamma} \\ 0 \\ 0 \end{bmatrix} \quad (8.67)$$

Furthermore, for a simplified analysis of these equations, one can assume linear relationships for these aerodynamic coefficients with respect to the trim sideslip angle, $\bar{\beta}$, trim aileron deflection $\bar{\delta}_a$, and trim rudder deflection, $\bar{\delta}_r$, i.e.

$$\bar{C}_S = C_{S\beta} \bar{\beta} + C_{S\delta_a} \bar{\delta}_a + C_{S\delta_r} \bar{\delta}_r \quad (8.68)$$

$$\bar{C}_l = C_{l\beta} \bar{\beta} + C_{l\delta_a} \bar{\delta}_a + C_{l\delta_r} \bar{\delta}_r \quad (8.69)$$

and

$$\bar{C}_n = C_{n\beta} \bar{\beta} + C_{n\delta_a} \bar{\delta}_a + C_{n\delta_r} \bar{\delta}_r \quad (8.70)$$

Then, by substitution, one has

$$\begin{bmatrix} Q_\infty S_w (C_{S\beta} \bar{\beta} + C_{S\delta_a} \bar{\delta}_a + C_{S\delta_r} \bar{\delta}_r) \cos \bar{\beta} - Q_\infty S_w \bar{C}_D \sin \bar{\beta} \\ Q_\infty S_w b_w (C_{l\beta} \bar{\beta} + C_{l\delta_a} \bar{\delta}_a + C_{l\delta_r} \bar{\delta}_r) \\ Q_\infty S_w b_w (C_{n\beta} \bar{\beta} + C_{n\delta_a} \bar{\delta}_a + C_{n\delta_r} \bar{\delta}_r) + \bar{N}_{p,S} \end{bmatrix} = \begin{bmatrix} -mg \sin \bar{\phi} \cos \bar{\gamma} \\ 0 \\ 0 \end{bmatrix} \quad (8.71)$$

Then, assuming $\bar{\beta}$ is small, i.e. $\sin \bar{\beta} \approx \bar{\beta}$ and $\cos \bar{\beta} \approx 1$, one has

$$\begin{bmatrix} \bar{C}_{S\beta} - \bar{C}_D & C_{S\delta_a} & C_{S\delta_r} \\ C_{l\beta} & C_{l\delta_a} & C_{l\delta_r} \\ C_{n\beta} & C_{n\delta_a} & C_{n\delta_r} \end{bmatrix} \begin{bmatrix} \bar{\beta} \\ \bar{\delta}_a \\ \bar{\delta}_r \end{bmatrix} = \begin{bmatrix} \frac{mg}{Q_\infty S_w} \cos \bar{\gamma} \sin \bar{\phi} \\ 0 \\ -\frac{\bar{N}_{p,S}}{Q_\infty S_w b_w} \end{bmatrix} \quad (8.72)$$

which can be solved by multiplying both sides by the inverse matrix on the left side. Notably, the maximum allowable roll angle, ϕ_{max} , plays a role in possible trim states for the airplane.

However, if one is required to balance the engine-out moment, $\bar{N}_{p,S}$ only using the rudder, then the required rudder deflection must be

$$\bar{\delta}_r = -\frac{\bar{N}_{p,S}}{Q_\infty S_w b_w C_{n\delta_r}} \quad (8.73)$$

which by setting the rudder deflection to its limit, $\bar{\delta}_r = \delta_{r,max}$, one has the **minimum control airspeed**, v_{mc} , i.e.

$$v_{mc} = \sqrt{-\frac{\bar{N}_{p,S}}{\frac{1}{2} \rho S_w b_w C_{n\delta_r} \delta_{r,max}}} \quad (8.74)$$

which is typically used in sizing the vertical tail and rudder.

References

For more information, please refer to the following

- Nelson, R. C., “3.5 Small-Disturbance Theory,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 104-108
- Schmidt, D. K., “9.1 Equilibrium Reference Conditions,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 481-485
- Schmidt, D. K., “9.3 Analysis of Steady Rectilinear Flight,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 500-523
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “2.3 Aircraft Forces and Moments,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 75-101
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “2.5 The Nonlinear Aircraft Model,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 108-116

8.3 Introductory Linearized Rigid Airplane Dynamics

This section details the linearization of a conventional airplane dynamics equations about straight flight in the stability frame with coordinated flight, i.e. $\bar{\beta} = 0$, and wings-level flight, i.e. $\bar{\phi} = 0$. Before beginning the derivation, it should be pointed out the airplane linearized, time-invariant state equation will use states $u, \alpha, \beta, p, q, r, \phi$, and θ , and inputs $\delta_a, \delta_e, \delta_r, T$ in perturbed form with leading Δ 's. The output equation will not be addressed in this section as it depends on the use of the dynamics model.

First, recall the airplane 6-DOF equations of motion as

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{u} + qu \sin \alpha - ru \tan \beta \\ \dot{u} \tan \beta + \dot{\beta} u \sec^2 \beta + ru - pu \sin \alpha \\ \dot{u} \sin \alpha + \dot{\alpha} u \cos \alpha + pu \tan \beta - qu \\ \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \quad (8.75)$$

Next, for straight, coordinated, wings-level steady-flight, i.e. $\dot{\psi} = \bar{\beta} = \bar{\phi} = 0$, one can decouple these into the linearized longitudinal and lateral-directional EOMs using the trim and perturbed states, forces, and moments as

$$\begin{bmatrix} \bar{X} + \Delta X - g \sin (\bar{\theta} + \Delta \theta) \\ \bar{Z} + \Delta Z + g \cos (\bar{\theta} + \Delta \theta) \cos \Delta \phi \\ \bar{M} + \Delta M \end{bmatrix} = \begin{bmatrix} \Delta \dot{u} + \Delta q (\bar{u} + \Delta u) \sin \Delta \alpha \\ \Delta \dot{u} \sin \Delta \alpha + \Delta \dot{\alpha} (\bar{u} + \Delta u) \cos \Delta \alpha - \Delta q (\bar{u} + \Delta u) \\ \Delta \dot{q} \end{bmatrix} \quad (8.76)$$

and

$$\begin{bmatrix} \bar{Y} + \Delta Y + g \cos \bar{\theta} \sin \Delta \phi \\ \bar{L} + \Delta L \\ \bar{N} + \Delta N \end{bmatrix} = \begin{bmatrix} \Delta \dot{\beta} \bar{u} \sec^2 \Delta \beta + \bar{u} (\Delta r) \\ \Delta \dot{p} - \frac{I_{xz}}{I_{xx}} \Delta \dot{r} \\ \Delta \dot{r} - \frac{I_{xz}}{I_{zz}} \Delta \dot{p} \end{bmatrix} \quad (8.77)$$

Then, using the trigonometric addition formulas and the small angle approximation, i.e.

$$\sin(\bar{a} + \Delta a) = \sin \bar{a} \cos \Delta a + \cos \bar{a} \sin \Delta a = \sin \bar{a} + \cos \bar{a} \Delta a \quad (8.78)$$

and

$$\cos(\bar{a} + \Delta a) = \cos \bar{a} \cos \Delta a - \sin \bar{a} \sin \Delta a = \cos \bar{a} - \sin \bar{a} \Delta a \quad (8.79)$$

for θ, ϕ, α , and β where $\bar{\phi} = \bar{\alpha} = \bar{\beta} = 0$ has already been assumed, one has

$$\begin{bmatrix} \bar{X} + \Delta X - g \sin \bar{\theta} - g \cos \bar{\theta} \Delta \theta \\ \bar{Z} + \Delta Z - g \sin \bar{\theta} \Delta \theta \\ \bar{M} + \Delta M \end{bmatrix} = \begin{bmatrix} \Delta \dot{u} + \Delta q (\bar{u} + \Delta u) \Delta \alpha \\ \Delta \dot{u} \Delta \alpha + \Delta \dot{\alpha} (\bar{u} + \Delta u) - \Delta q (\bar{u} + \Delta u) \\ \Delta \dot{q} \end{bmatrix} \quad (8.80)$$

and

$$\begin{bmatrix} \bar{Y} + \Delta Y - g \cos \bar{\theta} \Delta \phi \\ \bar{L} + \Delta L \\ \bar{N} + \Delta N \end{bmatrix} = \begin{bmatrix} \Delta \dot{\beta} \bar{u} + \Delta r \bar{u} \\ \Delta \dot{p} - \frac{I_{xz}}{I_{xx}} \Delta \dot{r} \\ \Delta \dot{r} - \frac{I_{xz}}{I_{zz}} \Delta \dot{p} \end{bmatrix} \quad (8.81)$$

These can be further linearized by eliminating higher-order terms of the perturbations and separating out the perturbation terms, i.e.

$$\begin{bmatrix} \bar{X} - g \sin \bar{\theta} \\ \bar{Z} \\ \bar{M} \end{bmatrix} + \begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \\ -g \sin \bar{\theta} \\ 0 \end{bmatrix} \Delta \theta = \begin{bmatrix} \Delta \dot{u} \\ \bar{u} \Delta \dot{\alpha} - \bar{u} \Delta q \\ \Delta \dot{q} \end{bmatrix} \quad (8.82)$$

and

$$\begin{bmatrix} \bar{Y} \\ \bar{L} \\ \bar{N} \end{bmatrix} + \begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \\ 0 \\ 0 \end{bmatrix} \Delta \phi = \begin{bmatrix} \Delta \dot{\beta} \bar{u} + \Delta r \bar{u} \\ \Delta \dot{p} - \frac{I_{xz}}{I_{xx}} \Delta \dot{r} \\ \Delta \dot{r} - \frac{I_{xz}}{I_{zz}} \Delta \dot{p} \end{bmatrix} \quad (8.83)$$

Next, by definition of straight steady-flight conditions for which

$$\begin{aligned} \bar{X} - g \sin \bar{\theta} &= 0 \\ \bar{Y} &= 0 \\ \bar{Z} + g \cos \bar{\theta} &= 0 \\ \bar{L} &= 0 \\ \bar{M} &= 0 \\ \bar{N} &= 0 \end{aligned} \quad (8.84)$$

due to $\vec{\omega}_{S/N} = 0$, the linearized models can be rewritten using matrices and the states as

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \\ -g \sin \bar{\theta} \\ 0 \end{bmatrix} \Delta \theta + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \bar{u} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{u} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} \quad (8.85)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} + \begin{bmatrix} -g \cos \bar{\theta} \\ 0 \\ 0 \end{bmatrix} \Delta \phi - \begin{bmatrix} 0 & 0 & \bar{u} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} = \begin{bmatrix} \bar{u} & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \end{bmatrix} \quad (8.86)$$

As a general airplane modeling principle, the perturbed aerodynamic and propulsive forces and moments are modeled as two sets

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} X_{\dot{u}} & X_{\dot{\alpha}} & X_{\dot{q}} \\ Z_{\dot{u}} & Z_{\dot{\alpha}} & Z_{\dot{q}} \\ M_{\dot{u}} & M_{\dot{\alpha}} & M_{\dot{q}} \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_\alpha & X_q \\ Z_u & Z_\alpha & Z_q \\ M_u & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_T \\ Z_{\delta_e} & Z_T \\ M_{\delta_e} & M_T \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta T \end{bmatrix} \quad (8.87)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} = \begin{bmatrix} Y_{\dot{\beta}} & Y_{\dot{p}} & Y_{\dot{r}} \\ L_{\dot{\beta}} & L_{\dot{p}} & L_{\dot{r}} \\ N_{\dot{\beta}} & N_{\dot{p}} & N_{\dot{r}} \end{bmatrix} \begin{bmatrix} \Delta \dot{\beta} \\ \Delta \dot{p} \\ \Delta \dot{r} \end{bmatrix} + \begin{bmatrix} Y_\beta & Y_p & Y_r \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (8.88)$$

where the coefficients of the perturbed states and inputs inside these matrices are called the **stability and control derivatives** and correspond to the Jacobian partial derivative terms about trimmed steady-flight.

As these derivatives generally change with an airplane's trim conditions, one typically calculates tables of these derivatives at many steady-flight conditions using wind tunnels tests, flight tests, and/or computational fluid dynamics (CFD). Methods for determining these derivatives from such data fall under the discipline of **airplane system identification**, in particular, optimal parameter estimation, a topic addressed later in this textbook. In this textbook, the linearized dynamics derivation for airplanes will assume the following stability and control derivatives dominate the perturbed forces and moments

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_\alpha & 0 \\ Z_u & Z_\alpha & Z_q \\ M_u & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} 0 & X_T \\ Z_{\delta_e} & Z_T \\ M_{\delta_e} & M_T \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta T \end{bmatrix} \quad (8.89)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} = \begin{bmatrix} Y_\beta & Y_p & Y_r \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (8.90)$$

Thus, substituting for the perturbed aerodynamic and propulsive forces and moments as assumed in this course, one has

$$\begin{aligned} & \begin{bmatrix} 0 & 0 & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_\alpha & 0 \\ Z_u & Z_\alpha & Z_q \\ M_u & M_\alpha & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} 0 & X_T \\ Z_{\delta_e} & Z_T \\ M_{\delta_e} & M_T \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta T \end{bmatrix} \\ & + \begin{bmatrix} -g \cos \bar{\theta} \\ -g \sin \bar{\theta} \\ 0 \end{bmatrix} \Delta \theta + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \bar{u} \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{u} & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} \end{aligned} \quad (8.91)$$

and

$$\begin{aligned} & \begin{bmatrix} Y_\beta & Y_p & Y_r \\ L_\beta & L_p & L_r \\ N_\beta & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \\ \Delta\theta \end{bmatrix} \\ & + \begin{bmatrix} -g \cos \bar{\theta} \\ 0 \\ 0 \end{bmatrix} \Delta\phi - \begin{bmatrix} 0 & 0 & \bar{u} \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \end{bmatrix} = \begin{bmatrix} \bar{u} & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 \end{bmatrix} \begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \end{bmatrix} \end{aligned} \quad (8.92)$$

Next, combining matrices of similar terms and reversing the sides of the equations, one has

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & \bar{u} - Z_\alpha & 0 \\ 0 & -M_\alpha & 1 \end{bmatrix} \begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \cos \bar{\theta} \\ Z_u & Z_\alpha & \bar{u} + Z_q & -g \sin \bar{\theta} \\ M_u & M_\alpha & M_q & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} 0 & X_T \\ Z_{\delta_e} & Z_T \\ M_{\delta_e} & M_T \end{bmatrix} \begin{bmatrix} \Delta\delta_e \\ \Delta T \end{bmatrix} \quad (8.93)$$

and

$$\begin{bmatrix} \bar{u} & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 \end{bmatrix} \begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \end{bmatrix} = \begin{bmatrix} Y_\beta & Y_p & Y_r - \bar{u} & -g \cos \bar{\theta} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \\ \Delta\theta \end{bmatrix} \quad (8.94)$$

Then, recalling for small angles

$$\Delta\dot{\theta} = \Delta q \quad (8.95)$$

and

$$\Delta\dot{\phi} = \Delta p + \tan \bar{\theta} \Delta r \quad (8.96)$$

one has

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \bar{u} - Z_\alpha & 0 & 0 \\ 0 & -M_\alpha & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \cos \bar{\theta} \\ Z_u & Z_\alpha & \bar{u} + Z_q & -g \sin \bar{\theta} \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} + \begin{bmatrix} 0 & X_T \\ Z_{\delta_e} & Z_T \\ M_{\delta_e} & M_T \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_e \\ \Delta T \end{bmatrix} \quad (8.97)$$

and

$$\begin{bmatrix} \bar{u} & 0 & 0 & 0 \\ 0 & 1 & -\frac{I_{xz}}{I_{xx}} & 0 \\ 0 & -\frac{I_{xz}}{I_{xx}} & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} Y_\beta & Y_p & Y_r - \bar{u} & -g \cos \bar{\theta} \\ L_\beta & L_p & L_r & 0 \\ N_\beta & N_p & N_r & 0 \\ 0 & 1 & \tan \bar{\theta} & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + \begin{bmatrix} 0 & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \\ \Delta\theta \end{bmatrix} \quad (8.98)$$

Finally, taking the inverse matrices of the left side, one has the **linearized longitudinal rigid airplane dynamics**

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \cos \bar{\theta} \\ \frac{Z_u}{\bar{u}-Z_\alpha} & \frac{Z_\alpha}{\bar{u}-Z_\alpha} & \frac{\bar{u}+Z_q}{\bar{u}-Z_\alpha} & -\frac{g}{\bar{u}-Z_\alpha} \sin \bar{\theta} \\ M_u + M_{\dot{\alpha}} \frac{Z_u}{\bar{u}-Z_\alpha} & M_\alpha + M_{\dot{\alpha}} \frac{Z_\alpha}{\bar{u}-Z_\alpha} & M_q + M_{\dot{\alpha}} \frac{\bar{u}+Z_q}{\bar{u}-Z_\alpha} & -M_{\dot{\alpha}} \frac{g}{\bar{u}-Z_\alpha} \sin \bar{\theta} \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} 0 & X_T \\ \frac{Z_{\delta_e}}{\bar{u}-Z_\alpha} & \frac{Z_T}{\bar{u}-Z_\alpha} \\ M_{\delta_e} + M_{\dot{\alpha}} \frac{Z_{\delta_e}}{\bar{u}-Z_\alpha} & M_T + M_{\dot{\alpha}} \frac{Z_T}{\bar{u}-Z_\alpha} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta T \end{bmatrix} \quad (8.99)$$

and one has the **linearized lateral-directional rigid airplane dynamics**

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{Y_\beta}{\bar{u}} & \frac{Y_p}{\bar{u}} & \frac{Y_r}{\bar{u}} - 1 & \frac{g}{\bar{u}} \cos \bar{\theta} \\ L_\beta^* & L_p^* & L_r^* & 0 \\ N_\beta^* & N_p^* & N_r^* & 0 \\ 0 & 1 & \tan \bar{\theta} & 0 \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \\ \Delta \phi \end{bmatrix} + \begin{bmatrix} 0 & \frac{Y_{\delta_r}}{\bar{u}} \\ L_{\delta_a}^* & L_{\delta_r}^* \\ N_{\delta_a}^* & N_{\delta_r}^* \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} \quad (8.100)$$

where

$$L_\bullet^* = \frac{L_\bullet + N_\bullet \frac{I_{xz}}{I_{zz}}}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \quad (8.101)$$

and

$$N_\bullet^* = \frac{N_\bullet + L_\bullet \frac{I_{xz}}{I_{zz}}}{1 - \frac{I_{xz}^2}{I_{xx} I_{zz}}} \quad (8.102)$$

for $\bullet = \beta, p, r, \delta_a$, and δ_r due to the coupling of L and N for non-zero I_{xz} . Furthermore, note that if $I_{xz} = 0$, then $L_\bullet^* = L_\bullet$ and $N_\bullet^* = N_\bullet$.

First, note that this derivation results in two sets of fourth-order LTI state-space systems. Secondly, it should be noted that one may also form these EOMs using the v and w states instead of β and α , respectively. Third, it should also be noted that often these EOMs are extended to fifth-order systems by including the perturbed altitude, Δh , as a perturbed longitudinal state and the perturbed yaw, $\Delta\psi$, as a perturbed lateral-directional state which are simply related to the other states for wings-level, coordinated steady-flight by

$$\Delta\dot{h} = \sin \bar{\theta} \Delta u + \bar{u} \cos \bar{\theta} (\Delta\theta - \Delta\alpha) \quad (8.103)$$

and

$$\Delta\dot{\psi} = \sec \bar{\theta} \Delta r \quad (8.104)$$

and do not have any direct control derivatives. Lastly, note that these derived state-space systems do not have an explicit output equation whose form will depend on which states should be used as “outputs,” e.g. in a control system.

Another important output that is sometimes used in the output equation is the acceleration vector at some position along the rigid airplane, $\vec{a}_p = [x_p \ y_p \ z_p]$ in stability frame coordinates, for which, in general

$$\vec{a}_p = \dot{\vec{v}}_S + [\vec{\omega}_{S/N}] \times \vec{v}_S + [\vec{a}_{S/N}] \times \vec{p}_S + [\vec{\omega}_{S/N}] \times [\vec{\omega}_{S/N}] \times \vec{p}_S \quad (8.105)$$

However, after linearizing and for $\bar{\beta} = 0$ and $\bar{\phi} = 0$, one has

$$\begin{bmatrix} \Delta a_{p,x} \\ \Delta a_{p,y} \\ \Delta a_{p,z} \end{bmatrix} = \begin{bmatrix} \Delta \dot{u} + \bar{w} \Delta q + z_p \Delta \dot{q} - y_p \Delta \dot{r} \\ \Delta \dot{v} + \bar{u} \Delta r - \bar{w} \Delta p + x_p \Delta \dot{r} - z_p \Delta \dot{p} \\ \Delta \dot{w} - \bar{u} \Delta q + y_p \Delta \dot{p} - x_p \Delta \dot{q} \end{bmatrix} \quad (8.106)$$

which can be used for the output equation for an accelerometer placed at \vec{p} .

Lastly, X , Y , Z , L_{roll} , M , and N can be written with aerodynamic coefficients for airplanes as

$$X = \frac{Q_\infty S_w}{m} C_X \quad (8.107)$$

$$Y = \frac{Q_\infty S_w}{m} C_Y \quad (8.108)$$

$$Z = \frac{Q_\infty S_w}{m} C_Z \quad (8.109)$$

$$L_{roll} = \frac{Q_\infty S_w b_w}{I_{xx}} C_l \quad (8.110)$$

$$M = \frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_m \quad (8.111)$$

and

$$N = \frac{Q_\infty S_w b_w}{I_{zz}} C_n \quad (8.112)$$

where lowercase letters are used for L , M , and N coefficients as C_L is already used for the lift coefficient.

All the aerodynamic coefficients in this section can be modeled using aircraft system identification (SID) via analytical equations, e.g. the **component build-up model** for conventional airplanes presented in appendix A, or estimated using wind tunnel and/or flight test data. Thus, the overall aircraft aerodynamic forces and moments are not simple to compute for arbitrary flight conditions as the aerodynamic coefficients are generally functions of the local airspeed, angle of attack, and sideslip angle at each lifting surface, their geometric layout, as well as the control inputs, and the linear and angular velocities of the vehicle. Thus, one typically linearizes the rigid aircraft dynamics about equilibrium flight conditions in order to analyze its response characteristics and design suitable control systems.

References

For more information, please refer to the following

- Nelson, R. C., “3.5 Small-Disturbance Theory,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 104-108
- Schmidt, D. K., “10.2 Linear Flight-Dynamics Perturbation,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 481-485
- Schmidt, D. K., “10.3 Decoupled Longitudinal and Lateral-Directional Linear Models,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 500-523
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “2.6 Linear Models and the Stability Derivatives,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 116-137

8.4 Airplane Stability and Flying Qualities

Longitudinal Modes and Stability

The fourth-order linearized longitudinal dynamics for rigid airplanes in straight-and-level flight as

$$\begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & 0 & -g \\ \frac{Z_u}{\bar{u}-Z_{\dot{\alpha}}} & \frac{Z_\alpha}{\bar{u}-Z_{\dot{\alpha}}} & \frac{\bar{u}+Z_q}{\bar{u}-Z_{\dot{\alpha}}} & 0 \\ M_u + M_{\dot{\alpha}} \frac{Z_u}{\bar{u}-Z_{\dot{\alpha}}} & M_\alpha + M_{\dot{\alpha}} \frac{Z_\alpha}{\bar{u}-Z_{\dot{\alpha}}} & M_q + M_{\dot{\alpha}} \frac{\bar{u}+Z_q}{\bar{u}-Z_{\dot{\alpha}}} & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} 0 & X_{\delta_T} \\ \frac{Z_{\delta_e}}{\bar{u}-Z_{\dot{\alpha}}} & \frac{Z_{\delta_T}}{\bar{u}-Z_{\dot{\alpha}}} \\ M_{\delta_e} + M_{\dot{\alpha}} \frac{Z_{\delta_e}}{\bar{u}-Z_{\dot{\alpha}}} & M_{\delta_T} + M_{\dot{\alpha}} \frac{Z_{\delta_T}}{\bar{u}-Z_{\dot{\alpha}}} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_T \end{bmatrix} \quad (8.113)$$

where $\Delta w \approx \bar{u}\Delta\alpha$ may also be substituted with

$$M_\alpha = \bar{u}M_w, \quad Z_\alpha = \bar{u}Z_w, \quad M_{\dot{\alpha}} = \bar{u}M_{\dot{w}} \quad (8.114)$$

There are conventionally two oscillatory modes that dominate the response in the longitudinal plane: the **short-period mode** and the **phugoid mode** derived from the Greek words, *phuge* and *eidos*, for “flight-like.” The phugoid mode is also known as the **long-period mode** as it is typically oscillatory though it may separate into two real modes for some airplanes. Recall from Lyapunov stability theory, an airplane is **longitudinally stable** if and only if both modes are stable which occurs when the real part of the eigenvalues of the state matrix A are negative, i.e. the modes are in the left-half plane (LHP) of the complex plane.

The short-period mode can be approximated by assuming $\Delta u = 0$. Then, one has the second-order **short-period mode approximation** as the characteristic polynomial

$$\phi_{S-P}(\lambda) = \lambda^2 + \left(-\frac{Z_\alpha}{\bar{u} - Z_{\dot{\alpha}}} - M_q - M_{\dot{\alpha}} \frac{\bar{u} + Z_q}{\bar{u} - Z_{\dot{\alpha}}} \right) \lambda + \left(M_q \frac{Z_\alpha}{\bar{u} - Z_{\dot{\alpha}}} - M_\alpha \frac{\bar{u} + Z_q}{\bar{u} - Z_{\dot{\alpha}}} \right) \quad (8.115)$$

The short-period mode primarily affects $\Delta\alpha$ and the short-term behavior of $\Delta\theta$. For airplane design, one typically balances a desire for a high natural frequency, i.e. a quick response to input, but also heavy damping, i.e. little overshoot. Increasing the short-period damping is done through a SAS called a **pitch damper**.

The phugoid mode can be approximated by assuming $\Delta\dot{\alpha} = \Delta\dot{q} = 0$. Then, one has the second-order **phugoid mode approximation** as the characteristic polynomial

$$\phi_{L-P}(\lambda) = \lambda^2 + \left(-X_u + X_\alpha \frac{Z_u M_q - M_u (\bar{u} + Z_q)}{Z_\alpha M_q - M_\alpha (\bar{u} + Z_q)} \right) \lambda + \left(g \frac{Z_u M_\alpha - Z_\alpha M_u}{Z_\alpha M_q - M_\alpha (\bar{u} + Z_q)} \right) \quad (8.116)$$

The phugoid mode primarily affects Δu and the long-term behavior of $\Delta\theta$ or Δh . Because the phugoid mode is typically highly oscillatory, most applications require it to be corrected during flight, often manually, but these corrections can be fatiguing for pilots if the damping ratio is too low. In this case, one can use a SAS using pitch feedback. Lastly, it should be noted that phugoid mode can be considered as the gradual interchange of kinetic and potential energy through the varying velocity and altitude, respectively.

Lateral-Directional Modes and Stability

The fourth-order linearized lateral-directional dynamics for rigid airplanes in straight-and-level flight as

$$\begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} = \begin{bmatrix} \frac{Y_\beta}{\bar{u}} & \frac{Y_p}{\bar{u}} & \frac{Y_r}{\bar{u}} - 1 & \frac{g}{\bar{u}} \cos \bar{\theta} \\ L_\beta^* & L_p^* & L_r^* & 0 \\ N_\beta^* & N_p^* & N_r^* & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} \quad (8.117)$$

$$+ \begin{bmatrix} 0 & \frac{Y_{\delta r}}{\bar{u}} \\ L_{\delta a}^* & L_{\delta r}^* \\ N_{\delta a}^* & N_{\delta r}^* \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix}$$

where $\Delta v \approx \bar{u}\Delta\beta$ may also be substituted with

$$Y_\beta = \bar{u}Y_v, \quad L_\beta^* = \bar{u}L_v^*, \quad N_\beta^* = \bar{u}N_v^* \quad (8.118)$$

There are conventionally two exponentially decaying modes and one oscillatory mode that dominate the responses in the lateral-directional planes: the **roll mode**, the **spiral mode**, and the **dutch-roll mode**. However, for some airplanes the **roll-spiral mode** may combine into a single oscillatory mode. The motion corresponding to the dutch-roll mode was named for its side-to-side motion which alludes to dutch figure skaters using repetitive right-and-left skating on the outer edge of their skates to maintain speed. Recall from Lyapunov stability theory, an airplane is **lateral-directionally stable** if and only if all three modes are stable which occurs when the real part of the eigenvalues of the state matrix A are negative, i.e. the modes are in the left-half plane (LHP) of the complex plane.

The roll mode can be approximated by assuming a pure rolling equation of motion. Then, one has the first-order **roll mode approximation** as the characteristic polynomial

$$\phi_R(\lambda) = \lambda - L_p^* \quad (8.119)$$

The roll mode primarily affects the short-term behavior of p . The value of L_p depends primarily on the size of the wing and tail surfaces. Typically this mode decays rapidly compared to the other lateral-directional modes, albeit sometimes too quickly. Reducing the roll mode time constant is done through a SAS called a **roll damper**.

The dutch-roll mode can be approximated by assuming $Y_p = Y_r = L_r^* = N_p^* = g/\bar{u} = 0$. Then, one has the second-order **dutch-roll mode approximation** as the characteristic polynomial

$$\phi_{D-R}(\lambda) = \lambda^2 + \left(-N_r^* - \frac{Y_\beta}{\bar{u}}\right)\lambda + \left(N_\beta^* + N_r^* \frac{Y_\beta}{\bar{u}}\right) \quad (8.120)$$

The dutch-roll mode primarily affects $\Delta\beta$, the long-term behavior of p , and the short-term behavior of $\Delta\phi$. For airplane design, one typically balances a desire for a high natural frequency, i.e. a quick response to input, but also heavy damping, i.e. little overshoot. Increasing the dutch-roll damping is done through a SAS called a **yaw damper**. In addition, a roll damper can be used in parallel to increase the effectiveness of the yaw damper.

The spiral mode can be approximated by assuming $\dot{\beta} = \dot{p} = \dot{r} = Y_p = Y_r = 0$. Then, one has the first-order **spiral mode approximation** as

$$\phi_S(\lambda) = \lambda + g \frac{L_\beta^* N_r^* - L_r^* N_\beta^*}{Y_\beta (L_r^* N_p^* - N_r^* L_p^*) + \bar{u} (L_\beta^* N_p^* - N_\beta^* L_p^*)} \quad (8.121)$$

The spiral mode primarily affects the long-term behavior of $\Delta\phi$. Typically this mode decays slowly compared to the other lateral-directional modes and may even be unstable, requiring the pilot to make corrections or a SAS added to the control system.

Lastly, it should be noted that airplane design characteristics affect the spiral and dutch-roll modes in opposite ways. In particular, increasing the dihedral effect makes the dutch-roll mode less stable and the spiral mode more stable. Conversely, increasing the directional stability makes the dutch-roll mode more stable and the spiral mode less stable. Thus, often one must use roll/yaw dampers to sufficiently stabilize both the spiral and dutch roll modes.

Flying Qualities

When designing aircraft, the **flying qualities**, also known as the **handling qualities**, i.e. the handling of the aircraft by a pilot, are closely related to the modal characteristics of the aircraft. However, as pilots are generally charged with performing various tasks or **missions**, the flying qualities are generally specified by airplane pilots according to the following three subjective levels:

- **Level 1** (Good): Flying qualities clearly adequate for the mission flight phase.
- **Level 2** (Acceptable): Flying qualities adequate to accomplish the mission flight phase, but with some increase in pilot workload and/or degradation in mission effectiveness or both.
- **Level 3** (Poor): Flying qualities such that the airplane can be controlled safely, but pilot workload is excessive and/or mission effectiveness is inadequate or both.

which also depend on the three generalized flight phase categories:

- **Category A:** nonterminal flight phases that require rapid maneuvering, precision tracking, or highly accurate flight-path control.
- **Category B:** nonterminal flight phases that are normally accomplished using gradual maneuvers and without precision tracking, although accurate flight-path control may be required.
- **Category C:** terminal flight phases that are normally accomplished using gradual maneuvers and usually require accurate flight-path control.

Notably, for Level 3 flying qualities, Category A flight phases can be terminated safely and Category B and C flight phases can be completed.

Lastly, another important part of assessing these flying qualities is by class of aircraft. For airplanes these are generally classified as

- **Class I:** Small, light airplanes
- **Class II:** Medium-weight, low-to-medium maneuverability airplanes
- **Class III:** Large, heavy, low-to-medium maneuverability airplanes
- **Class IV:** High-maneuverability airplanes

The following table of modal characteristics for different flying quality levels and flight phase categories for different classes of airplanes.

| Mode | Level | Category | Class | Characteristic(s) |
|--------------|-------|----------|-------------|---|
| Phugoid | 1 | All | All | $\zeta > 0.04$ |
| | 2 | All | All | $\zeta > 0$ |
| | 3 | All | All | $T > 55s$ |
| Short-Period | 1 | A and C | All | $0.35 \leq \zeta \leq 1.3$ |
| | | B | All | $0.3 \leq \zeta \leq 2.0$ |
| | 2 | A and C | All | $0.25 \leq \zeta \leq 2.0$ |
| | | B | All | $0.2 \leq \zeta \leq 2.0$ |
| | 3 | All | All | $0.15 \leq \zeta$ |
| | | | | |
| Roll | 1 | A and C | I, IV | $\tau \leq 1.0$ sec |
| | | | II, III | $\tau \leq 1.4$ sec |
| | | B | All | $\tau \leq 1.4$ sec |
| | 2 | A and C | I, IV | $\tau \leq 1.4$ sec |
| | | | II, III | $\tau \leq 3.0$ sec |
| | 3 | B | All | $\tau \leq 3.0$ sec |
| | | All | All | $\tau \leq 10$ sec |
| Spiral | 1 | A | I, IV | Doubling amplitude ≥ 12 sec |
| | | | II, III | Doubling amplitude ≥ 20 sec |
| | | B and C | All | Doubling amplitude ≥ 20 sec |
| | 2 | All | All | Doubling amplitude ≥ 12 sec |
| | 3 | All | All | Doubling amplitude ≥ 4 sec |
| Dutch-Roll | 1 | A | I, IV | $\zeta\omega_n \geq 0.35$ rad/s, $\zeta \geq 0.19$, $\omega_n > 1.0$ rad/s |
| | | | II, III | $\zeta\omega_n \geq 0.35$ rad/s, $\zeta \geq 0.19$, $\omega_n > 0.4$ rad/s |
| | | B | All | $\zeta\omega_n \geq 0.15$ rad/s, $\zeta \geq 0.08$, $\omega_n > 0.4$ rad/s |
| | | C | I, II-C, IV | $\zeta\omega_n \geq 0.15$ rad/s, $\zeta \geq 0.08$, $\omega_n > 1.0$ rad/s |
| | | | II-L, III | $\zeta\omega_n \geq 0.15$ rad/s, $\zeta \geq 0.08$, $\omega_n > 0.4$ rad/s |
| | 2 | All | All | $\zeta\omega_n \geq 0.05$ rad/s, $\zeta \geq 0.02$, $\omega_n > 0.4$ rad/s |
| | 3 | All | All | $\zeta \geq 0.02$, $\omega_n \geq 0.4$ rad/s |

where C and L denote carrier- or land-based airplanes. Note that the spiral mode requirements are for the doubling of amplitude for a potentially *unstable* spiral mode.

Lastly, it is important to note that the **Cooper-Harper Rating Scale (CHRS)** is another standard used by pilots and flight test engineers to rate the flying qualities of the airplane design as a whole. The CHRS scale goes from 1 to 10 with lower numbers corresponding to better flying qualities. The description for each rating is shown in the following table.

| Pilot Rating | Aircraft Characteristic | Demand of Pilot | Overall Assessment |
|--------------|--|---|--------------------|
| 1 | Excellent, highly desirable | Pilot compensation not a factor for desired performance | Good |
| 2 | Good, negligible deficiencies | Pilot compensation not a factor for desired performance | Good |
| 3 | Fair, some mildly unpleasant deficiencies | Minimal pilot compensation required for desired performance | Good |
| 4 | Minor, but annoying deficiencies | Desired performance requires moderate pilot compensation | Acceptable |
| 5 | Moderately objectionable deficiencies | Adequate performance requires considerable pilot compensation | Acceptable |
| 6 | Very objectionable, but tolerable deficiencies | Adequate performance requires extensive pilot compensation | Acceptable |
| 7 | Major deficiencies | Adequate performance not attainable with maximum tolerable compensation | Poor |
| 8 | Major deficiencies | Considerable pilot compensation is required for control | Poor |
| 9 | Major deficiencies | Intense pilot compensation is required for control | Poor |
| 10 | Major deficiencies | Control will be lost during some portion of required operation | Unacceptable |

References

For more information, please refer to the following

- Nelson, R. C., “4.4 Stick-Fixed Longitudinal Motion,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 147-151
- Nelson, R. C., “4.5 Longitudinal Approximations,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 152-162
- Nelson, R. C., “4.7 Flying Qualities,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 164-168
- Nelson, R. C., “5.4 Lateral-Directional Equations of Motion,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 193-203
- Nelson, R. C., “5.5 Lateral Flying Qualities,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 203-204
- Schmidt, D. K., “10.2 Linear Flight-Dynamics Perturbation,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 481-485

- Schmidt, D. K., “10.3 Decoupled Longitudinal and Lateral-Directional Linear Models,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 500-523
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.2 Aircraft Rigid-Body Modes,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 257-274
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.3 The Handling-Qualities Requirements,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 274-287

8.5 Rigid Rotary-Wing Aircraft Dynamics

Symmetrically Rotating Mass Effect

The presence of a mass rotating about its center of mass, i.e. symmetrically rotating, can be shown to have no effect on the translation equations of motion of a vehicle. However, the presence of a mass rotating about its center of mass does have an effect on the rotation equations of motion of a vehicle as this rotation directly contributes to the total angular momentum of the vehicle. To show this, consider the angular momentum of the vehicle as

$$\begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} = \dot{\vec{H}}_N = \dot{\vec{H}}_B + \vec{\omega}_{B \leftarrow N} \times \vec{H}_B \quad (8.122)$$

where \vec{H}_B consists of two components, the rigid vehicle’s angular momentum (with the mass of the propeller disk) and the propeller’s angular momentum, i.e.

$$\vec{H}_B = \vec{H}_{rig,B} + \vec{H}_{prop,B} \quad (8.123)$$

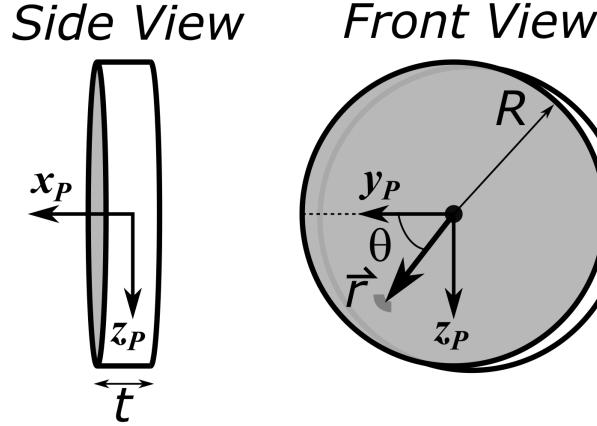
where

$$\vec{H}_{rig,B} = I_G \times \vec{\omega}_{B/N} \quad (8.124)$$

and the general form for the propeller angular momentum is

$$\vec{H}_{prop,B} = \int_{Vol} \vec{x}_B \times \dot{\vec{x}}_B \rho_V dV \quad (8.125)$$

where \vec{x}_B is the radial position of a mass element $\rho_V dV$ with respect to the center of mass. Next, one can idealize the propeller as a rotating disk with radius R and constant thickness t as



where the $x_P - y_P - z_P$ axes denotes the propeller-fixed frame centered at the disk's center (subscript P), the mass element volume at radius r is

$$dV = t(r d\theta) dr \quad (8.126)$$

and the disk has a radial mass distribution matching that of the propeller, i.e. setting the disk density, ρ_{disk} , to cover the entire disk volume, but with equivalent mass. Then, one can write the mass element velocity as

$$\dot{\vec{x}}_{B'} = \dot{\vec{x}}_P + \vec{\omega}_{B'/P} \times \vec{x}_P \quad (8.127)$$

where B' here is a body-fixed frame centered at the propeller with potentially different center and axes than B . Assuming the propeller disk is rigid, this can be rewritten as

$$\begin{aligned} \dot{\vec{x}}_{B'} &= \vec{\omega}_{B'/P} \times \vec{x}_P = \begin{bmatrix} \omega_{prop} \\ 0 \\ 0 \end{bmatrix} \times \begin{bmatrix} 0 \\ r \cos \theta \\ r \sin \theta \end{bmatrix} \\ &= \begin{bmatrix} 0 \\ \omega_{prop} r \sin \theta \\ -\omega_{prop} r \cos \theta \end{bmatrix} \end{aligned} \quad (8.128)$$

Thus, one has for the integrand of the angular velocity

$$\begin{aligned} \vec{x}_{B'} \times \rho_V \dot{\vec{x}}_{B'} &= \rho_V \begin{bmatrix} 0 \\ r \cos \theta \\ r \sin \theta \end{bmatrix} \times \begin{bmatrix} 0 \\ \omega_{prop} r \sin \theta \\ -\omega_{prop} r \cos \theta \end{bmatrix} \\ &= \begin{bmatrix} \omega_{prop} r^2 \\ 0 \\ 0 \end{bmatrix} \end{aligned} \quad (8.129)$$

and the angular momentum of the propeller disk in body frame coordinates is

$$\begin{aligned}\vec{H}_{prop,B'} &= \begin{bmatrix} \int_0^R \int_0^{2\pi} \omega_{prop} r^2 \rho_{disk} (r t d\theta dr) \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \omega_{prop} 2\pi t \int_0^R r^3 \rho_{disk} dr \\ 0 \\ 0 \end{bmatrix} \\ &= \begin{bmatrix} \omega_{prop} I_{prop} \\ 0 \\ 0 \end{bmatrix}\end{aligned}\quad (8.130)$$

where I_{prop} is the moment of inertia of the propeller about its center of mass.

Next, returning to the propeller angular momentum in the vehicle body frame centered at the center of mass, one has

$$\vec{H}_{prop,B} = \begin{bmatrix} h_{x,prop} \\ h_{y,prop} \\ h_{z,prop} \end{bmatrix} = C_{B \leftarrow B'} \begin{bmatrix} \omega_{prop} I_{prop} \\ 0 \\ 0 \end{bmatrix} \quad (8.131)$$

where the DCM, $C_{B \leftarrow B'}$, will depend on the orientation of the propeller relative to the body frame. For example, if the B' at some positive rotation angle, τ_P , about the y_B axis, then

$$\vec{H}_{prop,B} = \begin{bmatrix} \omega_{prop} I_{prop} \cos \tau_P \\ 0 \\ -\omega_{prop} I_{prop} \sin \tau_P \end{bmatrix} \quad (8.132)$$

Then, by substituting the additive angular momentum terms into the angular momentum differential equation, one has

$$\begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} = \left(\dot{\vec{H}}_{rig,B} + \dot{\vec{H}}_{prop,B} \right) + \vec{\omega}_{B \leftarrow N} \times \left(\vec{H}_{rig,B} + \vec{H}_{prop,B} \right) \quad (8.133)$$

$$\begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} = \left(\dot{\vec{H}}_{rig,B} + \vec{\omega}_{B \leftarrow N} \times \vec{H}_{rig,B} \right) + \left(\dot{\vec{H}}_{prop,B} + \vec{\omega}_{B \leftarrow N} \times \vec{H}_{prop,B} \right) \quad (8.134)$$

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} pr - \frac{I_{xy}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} + \begin{bmatrix} \frac{1}{I_{xx}} (\dot{h}_{x,prop} - rh_{y,prop} + qh_{z,prop}) \\ \frac{1}{I_{yy}} (\dot{h}_{y,prop} + rh_{x,prop} - ph_{z,prop}) \\ \frac{1}{I_{zz}} (\dot{h}_{z,prop} - qh_{x,prop} + ph_{y,prop}) \end{bmatrix} \quad (8.135)$$

Thus, the effect of adding additional rotating masses simply results in more additive terms which are a function of additional states, e.g. the propeller angular rate would be a control input that would affect the \dot{p} , \dot{q} and \dot{r} equations whether in the nonlinear state-space EOMs or in the LTI state-space EOMs. Note that additional rotating masses can easily be added to this model, though for an even number of rotating masses, often each is spun in opposite directions to counteract this contribution.

8.6 Axisymmetric Spacecraft Attitude Dynamics and Stability

Rotating Axisymmetric Spacecraft Dynamics in Torque-Free Motion

If one assumes $\vec{M}_{g,B} = 0$, i.e., the spacecraft is undergoing **torque-free motion**, one has

$$\begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} = \frac{d}{dt} \vec{H}_G = \begin{bmatrix} I_1 \dot{p} + (I_3 - I_2)qr \\ I_2 \dot{q} + (I_1 - I_3)pr \\ I_3 \dot{r} + (I_2 - I_1)pq \end{bmatrix} \quad (8.136)$$

Furthermore, by definition of the $3 - 1 - 3$ Euler angles, the nutation angle, θ_n , defines the angle between the z_B -axis and \vec{H}_G . Thus, one can state using the dot product

$$\cos \theta_n = \frac{\vec{H}_G \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}}{\|\vec{H}_G\|_2} = \frac{1}{\|\vec{H}_G\|_2} \begin{bmatrix} I_1 p \\ I_2 q \\ I_3 r \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (8.137)$$

Thus, one has

$$\cos \theta_n = \frac{1}{\|\vec{H}_G\|_2} I_3 r \quad (8.138)$$

or

$$\|\vec{H}_G\|_2 = \frac{I_3 r}{\cos \theta_n} \quad (8.139)$$

Taking the derivative with respect to time and assuming torque-free motion, one has

$$\frac{d \cos \theta_n}{dt} = \frac{1}{\|\vec{H}_G\|_2} \begin{bmatrix} I_1 p \\ I_2 q \\ I_3 r \end{bmatrix} \cdot \left([\vec{\omega}_{B/I,B}]_{\times} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \right) \quad (8.140)$$

$$-\sin \theta_n \dot{\theta}_n = \frac{1}{\|\vec{H}_G\|_2} \begin{bmatrix} I_1 p \\ I_2 q \\ I_3 r \end{bmatrix} \cdot \begin{bmatrix} q \\ -p \\ 0 \end{bmatrix} \quad (8.141)$$

or

$$\dot{\theta}_n = \omega_n = -\frac{(I_1 - I_2)pq}{\|\vec{H}_G\|_2 \sin \theta_n} \quad (8.142)$$

Thus, the first aspect to note for torque-free motion is that the nutation rate, ω_n , vanishes only if $I_1 = I_2$, i.e. the z_B -axis is an axis of “rotational symmetry” as the placement of the x_B - and y_B -axes can be switched for the principal body-fixed frame. If the x_B and y_B axes are arbitrary in the $x_B - y_B$ plane, then the spacecraft is said to be **axisymmetric** about z_B .

For this introductory look at torque-free motion, assume $I_1 = I_2$, then the torque-free motion equations become

$$\sum \vec{M}_B = \frac{d}{dt} \vec{H}_G = \begin{bmatrix} I_1 \dot{p} + (I_3 - I_1)qr \\ I_1 \dot{q} + (I_1 - I_3)pr \\ I_3 \dot{r} \end{bmatrix} = 0 \quad (8.143)$$

Thus, the spin rate is constant, i.e. $r = \bar{r}$ and

$$\|\vec{H}_G\|_2 = \frac{I_3 \bar{r}}{\cos \theta_n} \quad (8.144)$$

Next, define the *temporary angular rate*, ω_* , as

$$\omega_* = \frac{I_1 - I_3}{I_1} \bar{r} \quad (8.145)$$

which is related to another angular rate as will be shown.

This allows one to rewrite the torque-free motion dynamics as two coupled differential equations

$$\begin{aligned} \dot{p} - \omega_* q &= 0 \\ \dot{q} + \omega_* p &= 0 \end{aligned} \quad (8.146)$$

Taking the Laplace transform and separating variables, one has the eigenvalue equation

$$\begin{bmatrix} s & -\omega_* \\ \omega_* & s \end{bmatrix} \begin{bmatrix} p(s) \\ q(s) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (8.147)$$

with left-side determinant, or characteristic polynomial, as

$$s^2 + \omega_*^2 = 0 \quad (8.148)$$

which has two purely imaginary roots or system poles.

This equation has the well-known solution

$$\begin{aligned} p &= \Omega \sin \omega_* t \\ q &= \Omega \cos \omega_* t \end{aligned} \quad (8.149)$$

which corresponds to

$$\vec{\omega} = \begin{bmatrix} \Omega \sin \omega_* t \\ \Omega \cos \omega_* t \\ \bar{r} \end{bmatrix} \quad (8.150)$$

which describes the sweeping out of the **space cone** at constant nutation angle, θ_n , about \vec{H}_G and constant height, \bar{r} , above the $x_B - y_B$ plane with a circular base of radius Ω about the z_B -axis.

Next, consider the description of this motion using the 3 – 1 – 3 Euler angles. Substituting these expressions into the 3 – 1 – 3 Euler angle rate equations, i.e.

$$\begin{bmatrix} \omega_p \\ \omega_n \\ \omega_s \end{bmatrix} = \begin{bmatrix} \sin \psi_s \csc \theta_n & \cos \psi_s \csc \theta_n & 0 \\ \cos \psi_s & -\sin \psi_s & 0 \\ -\sin \psi_s \cot \theta_n & -\cos \psi_s \cot \theta_n & 1 \end{bmatrix} \begin{bmatrix} \Omega \sin \omega_* t \\ \Omega \cos \omega_* t \\ \bar{r} \end{bmatrix} \quad (8.151)$$

one has

$$\begin{bmatrix} \omega_p \\ \omega_n \\ \omega_s \end{bmatrix} = \begin{bmatrix} \Omega \sin \omega_* t \sin \psi_s \csc \theta_n + \Omega \cos \omega_* t \cos \psi_s \csc \theta_n \\ \Omega \sin \omega_* t \cos \psi_s - \Omega \cos \omega_* t \sin \psi_s \\ -\Omega \sin \omega_* t \sin \psi_s \cot \theta_n - \Omega \cos \omega_* t \cos \psi_s \cot \theta_n + \bar{r} \end{bmatrix} \quad (8.152)$$

By trigonometric identities, one has

$$\begin{bmatrix} \omega_p \\ \omega_n \\ \omega_s \end{bmatrix} = \begin{bmatrix} \Omega \csc \theta_n \cos(\omega_* t - \psi_s) \\ \Omega (\sin \omega_* t - \psi_s) \\ \bar{r} - \Omega \cot \theta_n \cos(\omega_* t - \psi_s) \end{bmatrix} \quad (8.153)$$

However, as $\omega_n = 0$ for $I_1 = I_2$, one has $\sin \omega_* t - \psi_s = 0$ and $\cos(\omega_* t - \psi_s) = 1$, which results in

$$\begin{bmatrix} \omega_p \\ \omega_n \\ \omega_s \end{bmatrix} = \begin{bmatrix} \Omega \csc \theta_n \\ 0 \\ \bar{r} - \Omega \cot \theta_n \end{bmatrix} \quad (8.154)$$

Furthermore, one must have

$$\omega_s = \dot{\psi}_s = \omega_* = \frac{I_1 - I_3}{I_1} \bar{r} \quad (8.155)$$

and

$$\vec{H}_{G,B} = \begin{bmatrix} I_1 \Omega \sin \omega_s t \\ I_1 \Omega \cos \omega_s t \\ I_3 \bar{r} \end{bmatrix} \quad (8.156)$$

However, by back-substitution into the third component yields

$$\Omega = \frac{I_3}{I_1} \bar{r} \tan \theta_n \quad (8.157)$$

Substituting for Ω into the first component of the angular velocity yields

$$\bar{r} = \frac{I_1}{I_3} \omega_p \cos \theta_n \quad (8.158)$$

and, substituting this into the angular momentum magnitude yields

$$\|\vec{H}_G\|_2 = I_1 \omega_p \quad (8.159)$$

Lastly, substituting for \bar{r} in the equation for ω_s , one has

$$\omega_s = \frac{I_1 - I_3}{I_3} \cos \theta_n \omega_p \quad (8.160)$$

Thus, for a **oblate body**, i.e. $I_1 < I_3$, then ω_p has the opposite sign as ω_s and one has a **retrograde precession**. For a **prolate body**, i.e. $I_1 > I_3$, then ω_p has the same sign as ω_s and one has a **prograde precession**. γ is the **wobble angle** defined as the angle between $\vec{\omega}_{B/I,B}$ and the z_B -axis, i.e.

$$\cos \gamma = \frac{r}{\|\vec{\omega}_{B/I,B}\|} = \frac{\bar{r}}{\sqrt{\Omega^2 \sin^2(\omega_* t) + \Omega^2 \cos^2(\omega_* t) + \bar{r}^2}} = \frac{\bar{r}}{\sqrt{\Omega^2 + \bar{r}^2}} \quad (8.161)$$

Substituting for Ω , one has

$$\cos \gamma = \frac{\omega_0}{\sqrt{\left(\frac{I_3}{I_1} \bar{r} \tan \theta_n\right)^2}} = \frac{I_1}{\sqrt{I_1^2 + I_3^2 \tan^2 \theta_n}} \quad (8.162)$$

and using trigonometric identities, one has

$$\cos \gamma = \frac{\cos \theta_n}{\sqrt{\frac{I_3^2}{I_1^2} + \left(1 - \frac{I_3^2}{I_1^2}\right) \cos^2 \theta_n}} \quad (8.163)$$

which shows that $\gamma > \theta$ for prograde precession and $\gamma < \theta$ for retrograde precession.

Stability of Single-Spinning Satellite

Consider the equilibrium condition where the angular velocity is about only the spin axis, i.e., without loss of generality, the z_B -axis. Thus, $r(t) = \bar{r}$ for all $t > 0$, $\bar{p} = 0^\circ$, and $\bar{q} = 0^\circ$, which trivially solves the torque-free equations of motion. Then, using the perturbation notation, the single-spinning spacecraft dynamics in torque-free motion are given by

$$\begin{bmatrix} I_1 \Delta \dot{p} + (I_3 - I_2) (\bar{r} + \Delta r) \Delta q \\ I_2 \Delta \dot{q} + (I_1 - I_3) (\bar{r} + \Delta r) \Delta p \\ I_3 \Delta \dot{r} + (I_2 - I_1) \Delta p \Delta q \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (8.164)$$

Then, keeping only the first-order terms, one has

$$\begin{bmatrix} I_1 \Delta \dot{p} + (I_3 - I_2) \bar{r} \Delta q \\ I_2 \Delta \dot{q} + (I_1 - I_3) \bar{r} \Delta p \\ I_3 \Delta \dot{r} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (8.165)$$

which implies $\Delta \dot{r} = \Delta \dot{r}_0$ is decoupled from the other perturbations.

Thus, one has

$$\begin{bmatrix} \Delta \dot{p} + \frac{(I_3 - I_2)}{I_1} \bar{r} \Delta q \\ \Delta \dot{q} + \frac{(I_1 - I_3)}{I_2} \bar{r} \Delta p \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (8.166)$$

Taking the Laplace transform and separating variables, one has the eigenvalue equation

$$\begin{bmatrix} s & \frac{(I_3 - I_2)}{I_1} \bar{r} \\ \frac{(I_1 - I_3)}{I_2} \bar{r} & s \end{bmatrix} \begin{bmatrix} \Delta p(s) \\ \Delta q(s) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (8.167)$$

with determinant, or characteristic equation, as

$$s^2 - \frac{(I_1 - I_3)(I_3 - I_2)}{I_1 I_2} \bar{r}^2 = 0 \quad (8.168)$$

Thus, by this analysis, if the spin axis is either the major axis, i.e. $I_3 > I_1$ and $I_3 > I_2$, or if the spin axis is the minor axis, i.e. $I_3 < I_1$ and $I_3 < I_2$, then the system poles are purely imaginary and the motion is sinusoidal and marginally stable as there is no damping in the torque-free motion model. However, if the spin axis is the intermediate axis, i.e. $I_1 > I_3 > I_2$ or $I_2 > I_3 > I_1$, then the motion is unstable due to a RHP system pole.

However, in reality, all spacecrafts experience some degree of flexibility which leads to additional stability analysis of the spin axis as the major or minor axis due to the energy dissipation into the structural vibrations. Consider a spacecraft with $I_1 = I_2$ and rotational kinetic energy

$$T_R = \frac{1}{2}I_1p^2 + \frac{1}{2}I_1q^2 + \frac{1}{2}I_3r^2 \quad (8.169)$$

Defining $\omega_{\perp}^2 = p^2 + q^2$ as the perpendicular angular momentum to the spin axis, one has

$$T_R = \frac{1}{2}I_1\omega_{\perp}^2 + \frac{1}{2}I_3r^2 \quad (8.170)$$

and differentiating with respect to time, one has

$$\dot{T}_R = \frac{1}{2}I_1 \frac{d\omega_{\perp}^2}{dt} + I_3r\dot{r} \quad (8.171)$$

where $\dot{T}_R < 0$ for all spacecrafts due to the energy dissipation.

Then, recall the angular momentum of a spacecraft with $I_1 = I_2$ is

$$\vec{H}_{G,B} = \begin{bmatrix} I_1p \\ I_1q \\ I_3r \end{bmatrix} \quad (8.172)$$

with magnitude

$$\|\vec{H}_G\|_2^2 = I_1^2(p^2 + q^2) + I_3^2r^2 = I_1^2\omega_{\perp}^2 + I_3^2r^2 \quad (8.173)$$

Differentiating the angular momentum magnitude with respect to time yields

$$\frac{d\|\vec{H}_G\|_2^2}{dt} = I_1^2 \frac{d\omega_{\perp}^2}{dt} + 2I_3^2r\dot{r} = 0 \quad (8.174)$$

which holds for torque-free motion. Thus, one has

$$\dot{r} = -\frac{I_1^2}{2I_3^2r} \frac{d\omega_{\perp}^2}{dt} \quad (8.175)$$

Finally, substituting this expression into the energy dissipation equation, one obtains

$$\dot{T}_R = \frac{1}{2}I_1 \frac{d\omega_{\perp}^2}{dt} + I_3r \left(-\frac{I_1^2}{2I_3^2r} \frac{d\omega_{\perp}^2}{dt} \right) \quad (8.176)$$

$$\dot{T}_R = \left(\frac{I_1I_3}{2I_3} - \frac{I_1^2}{2I_3} \right) \frac{d\omega_{\perp}^2}{dt} \quad (8.177)$$

and rearranging, one has

$$\frac{d\omega_{\perp}^2}{dt} = \frac{2I_3}{I_1(I_3 - I_1)} \dot{T}_R \quad (8.178)$$

This equation shows that if one has a oblate spacecraft, i.e. $I_3 > I_1$, then $\frac{d\omega_1^2}{dt} < 0$ and any perturbation in p or q will asymptotically decay to 0. Thus, the spin is asymptotically stable. However, if one has a prolate spacecraft, i.e. $I_3 < I_1$, then $\frac{d\omega_1^2}{dt} > 0$ and the spin is asymptotically unstable. Thus, if a spacecraft is to be **single-spin stabilized**, it must be an oblate spinner. Notably, a **nutation damper** can be used to passively increase this rate of energy dissipation for increased stability. These devices are made of viscous fluid and may or may not have a mass attached to springs.

References

For more information, please refer to the following

- Curtis, H. D., “9.4 Equations of Rotational Motion,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 420-414
- Curtis, H. D., “9.5 Moments of Inertia,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 414-435
- Curtis, H. D., “9.6 Euler’s Equations,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 435-440
- Curtis, H. D., “10.2 Torque-Free Motion,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 476-486
- Curtis, H. D., “10.3 Stability of Torque-Free Motion,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 486-491
- Hibbeler, R. C., “21.6 Torque-Free Motion,” *Dynamics*, 15th ed., Vol. 1, Pearson Inc., 2022, pp. 606-609
- Sidi, M. J., “4.5 Euler’s Moment Equations,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 95-98
- Sidi, M. J., “4.6 Characteristics of Rotational Motion of a Spinning Body,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 98-100

8.7 Gravity-Gradient Satellite Attitude Dynamics and Stability

Linearized Attitude Dynamics about Circular Orbit

Consider the satellite attitude equations of motion as

$$\vec{M}_B = \begin{bmatrix} I_{xx}\dot{p} + (I_{zz} - I_{yy})qr \\ I_{yy}\dot{q} + (I_{xx} - I_{zz})pr \\ I_{zz}\dot{r} + (I_{yy} - I_{xx})pq \end{bmatrix} \quad (8.179)$$

where, one can define the following for the inertial angular velocity

$$\vec{\omega}_{B/P,B} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \vec{\omega}_{B/L,B} + \vec{\omega}_{L/P,B} \quad (8.180)$$

where one the orbital angular velocity $\omega_O = -\frac{\mu}{a^3}$ for a circular orbit explicitly enters the dynamics equation via

$$\vec{\omega}_{L/P,L} = \begin{bmatrix} 0 \\ -\omega_O \\ 0 \end{bmatrix} \quad (8.181)$$

Thus, one has

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \vec{\omega}_{B/L,B} + C_{B \leftarrow L} \begin{bmatrix} 0 \\ -\omega_O \\ 0 \end{bmatrix} \quad (8.182)$$

which can be written in terms of the $3 - 2 - 1$ Euler angles of the body-fixed frame relative to the LVLH frame and the corresponding Euler angle rates as

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \phi \\ 0 & \cos \phi & \sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \cos \theta \cos \psi & \cos \theta \sin \psi & -\sin \theta \\ \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \sin \phi \cos \theta \\ \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} 0 \\ -\omega_O \\ 0 \end{bmatrix} \quad (8.183)$$

This angular velocity can be linearized for small Euler angles *and* small Euler angle rates using the small angle approximation which results in

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} \dot{\phi} - \phi \dot{\psi} \\ \dot{\theta} + \sin \phi \dot{\psi} \\ \dot{\psi} - \phi \dot{\theta} \end{bmatrix} + \begin{bmatrix} 1 & \psi & -\theta \\ -\psi & 1 & \phi \\ \theta & -\phi & 1 \end{bmatrix} \begin{bmatrix} 0 \\ -\omega_O \\ 0 \end{bmatrix} \quad (8.184)$$

and discarding higher-order terms, one has

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} \approx \begin{bmatrix} \dot{\phi} - \psi \omega_O \\ \dot{\theta} - \omega_O \\ \dot{\psi} + \phi \omega_O \end{bmatrix} \quad (8.185)$$

and, by differentiation,

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} \approx \begin{bmatrix} \ddot{\phi} - \omega_O \dot{\psi} \\ \ddot{\theta} \\ \ddot{\psi} + \omega_O \dot{\phi} \end{bmatrix} \quad (8.186)$$

Finally, by substitution, one has

$$\vec{M}_B = \begin{bmatrix} I_{xx}\ddot{\phi} - I_{xx}\omega_O\dot{\psi} + (I_{zz} - I_{yy})(\dot{\theta} - \omega_O)(\dot{\psi} + \phi\omega_O) \\ I_{yy}\ddot{\theta} + (I_{xx} - I_{zz})(\dot{\phi} - \psi\omega_O)(\dot{\psi} + \phi\omega_O) \\ I_{zz}\ddot{\psi} - I_{zz}\omega_O\dot{\phi} + (I_{yy} - I_{xx})(\dot{\phi} - \psi\omega_O)(\dot{\theta} - \omega_O) \end{bmatrix} \quad (8.187)$$

and discarding higher-order terms, one has

$$\vec{M}_B = \begin{bmatrix} I_{xx}\ddot{\phi} - \omega_O(I_{xx} + I_{zz} - I_{yy})\dot{\psi} + \omega_O^2(I_{zz} - I_{yy})\phi \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} + \omega_O(I_{xx} + I_{zz} - I_{yy})\dot{\phi} + \omega_O^2(I_{yy} - I_{zz})\psi \end{bmatrix} \quad (8.188)$$

Note that the linearized pitch dynamics are decoupled from the linearized roll and yaw dynamics.

For non-spinning satellites, one typically considers the effects of the gravity-gradient moment, \vec{M}_g , separate from other moment disturbances, \vec{M}_d , and control inputs, \vec{M}_c , which can be defined for a rigid-body as

$$\vec{M}_{g,B} = \int [\vec{x}_B] \times d\vec{F}_G = \int_V [\vec{x}_B] \times \left(-\frac{\mu\rho dV}{\|\vec{r}_m\|_2^3} \vec{r}_m \right) \quad (8.189)$$

where $dm = \rho dV$ is an infinitesimal mass element of the body with density ρ , $\vec{r}_m = \vec{r}_{P,L} + \vec{x}_B$ is the position vector of the mass element with respect to the origin of the perifocal frame, and $\vec{r}_{P,B}$ is the position of the center of mass of the rigid-body with respect to the origin of the perifocal frame expressed in body-fixed frame coordinates, i.e., for a circular orbit

$$\vec{r}_{P,B} = C_{B \leftarrow L} \begin{bmatrix} 0 \\ 0 \\ -a \end{bmatrix} = \begin{bmatrix} a \sin \theta \\ -a \sin \phi \cos \theta \\ -a \cos \phi \cos \theta \end{bmatrix} \quad (8.190)$$

With the assumption $\vec{x}_B \ll \vec{r}_P$, one can approximate the radial distance cubed by the truncated Taylor series

$$\frac{1}{\|\vec{r}_m\|_2^3} \approx \frac{1}{a^3} \left(1 - \frac{3\vec{r}_P \cdot \vec{x}_B}{a^2} \right) \quad (8.191)$$

which provides

$$\vec{M}_{g,B} = \frac{3\mu}{a^5} \int_V (\vec{r}_P \cdot \vec{x}_B) ([\vec{x}_B] \times \vec{r}_P) \rho dV \quad (8.192)$$

and by the use of the principal axes, all cross-products of inertia's are zero, which results in

$$\vec{M}_{g,B} \approx \frac{3\mu}{a^5} \begin{bmatrix} r_{P,y}r_{P,z} \left(\int y_B^2 \rho dV - \int z_B^2 \rho dV \right) \\ r_{P,x}r_{P,z} \left(\int z_B^2 \rho dV - \int x_B^2 \rho dV \right) \\ r_{P,x}r_{P,y} \left(\int x_B^2 \rho dV - \int y_B^2 \rho dV \right) \end{bmatrix} \quad (8.193)$$

Substituting for the moment of inertia integrals and the components of \vec{r}_P , one has

$$\vec{M}_{g,B} \approx \frac{3\mu}{a^5} \begin{bmatrix} (-a \sin \phi \cos \theta)(-a \cos \phi \cos \theta)(I_{zz} - I_{yy}) \\ (a \sin \theta)(-a \cos \phi \cos \theta)(I_{xx} - I_{zz}) \\ (a \sin \theta)(-a \sin \phi \cos \theta)(I_{yy} - I_{xx}) \end{bmatrix} \quad (8.194)$$

which simplifies to

$$\vec{M}_{g,B} \approx \frac{3\mu}{2a^3} \begin{bmatrix} (I_{zz} - I_{yy}) \sin(2\phi) \cos^2(\theta) \\ (I_{zz} - I_{xx}) \sin(2\theta) \cos \phi \\ (I_{xx} - I_{yy}) \sin(2\theta) \sin \phi \end{bmatrix} \quad (8.195)$$

Using the small-angle approximation, one has

$$\vec{M}_{g,B} = \frac{3\omega_O^2}{2} \begin{bmatrix} 2\phi(I_{zz} - I_{yy}) \\ 2\theta(I_{zz} - I_{xx}) \\ 2\theta\phi(I_{xx} - I_{yy}) \end{bmatrix} \quad (8.196)$$

and discarding higher-order terms, one has the **linearized gravity-gradient moment** vector

$$\vec{M}_{g,B} = \begin{bmatrix} 3\omega_O^2(I_{zz} - I_{yy})\phi \\ 3\omega_O^2(I_{zz} - I_{xx})\theta \\ 0 \end{bmatrix} \quad (8.197)$$

Thus, by substitution, for the **linearized gravity-gradient attitude dynamics**, one has

$$\begin{bmatrix} M_{d,x} + M_{c,x} \\ M_{d,y} + M_{c,y} \\ M_{d,z} + M_{c,z} \end{bmatrix} + \begin{bmatrix} 3\omega_O^2(I_{zz} - I_{yy})\phi \\ \omega_O^2(I_{zz} - I_{xx})\theta \\ 0 \end{bmatrix} = \begin{bmatrix} I_{xx}\ddot{\phi} - \omega_O(I_{xx} + I_{zz} - I_{yy})\dot{\psi} + \omega_O^2(I_{zz} - I_{yy})\phi \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} + \omega_O(I_{xx} + I_{zz} - I_{yy})\dot{\phi} + \omega_O^2(I_{yy} - I_{zz})\psi \end{bmatrix} \quad (8.198)$$

or

$$\begin{bmatrix} M_{d,x} + M_{c,x} \\ M_{d,y} + M_{c,y} \\ M_{d,z} + M_{c,z} \end{bmatrix} = \begin{bmatrix} I_{xx}\ddot{\phi} + 4\omega_O^2(I_{yy} - I_{zz})\phi - \omega_O(I_{xx} + I_{zz} - I_{yy})\dot{\psi} \\ I_{yy}\ddot{\theta} + 3\omega_O^2(I_{xx} - I_{zz})\theta \\ I_{zz}\ddot{\psi} + \omega_O^2(I_{yy} - I_{xx})\psi + \omega_O(I_{xx} + I_{zz} - I_{yy})\dot{\phi} \end{bmatrix} \quad (8.199)$$

Stability of Gravity-Gradient Satellite Dynamics

To analyze the stability of the linearized gravity-gradient attitude dynamics, first, define the moments of inertia ratios as

$$\sigma_{xx} = \frac{I_{yy} - I_{zz}}{I_{xx}} = \frac{\int(x^2 + z^2)dm - \int(x^2 + y^2)dm}{\int(y^2 + z^2)dm} = \frac{\int(z^2 - y^2)dm}{\int(z^2 + y^2)dm} \quad (8.200)$$

$$\sigma_{yy} = \frac{I_{xx} - I_{zz}}{I_{yy}} = \frac{\int(y^2 + z^2)dm - \int(x^2 + y^2)dm}{\int(x^2 + z^2)dm} = \frac{\int(z^2 - x^2)dm}{\int(z^2 + x^2)dm} \quad (8.201)$$

$$\sigma_{zz} = \frac{I_{yy} - I_{xx}}{I_{zz}} = \frac{\int(x^2 + z^2)dm - \int(y^2 + z^2)dm}{\int(x^2 + y^2)dm} = \frac{\int(x^2 - y^2)dm}{\int(x^2 + y^2)dm} \quad (8.202)$$

Thus,

$$|\sigma_{xx}| < 1 \quad \& \quad |\sigma_{yy}| < 1 \quad \& \quad |\sigma_{zz}| < 1 \quad (8.203)$$

Ignoring the disturbances, the linearized pitch state-space dynamics can be rearranged as

$$\begin{bmatrix} \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -3\omega_O^2\sigma_{yy} & 0 \end{bmatrix} \begin{bmatrix} \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ I_{yy}^{-1} \end{bmatrix} M_{c,y} \quad (8.204)$$

with characteristic equation

$$\lambda^2 + 3\omega_O^2\sigma_{yy} = 0 \quad (8.205)$$

and two roots

$$\lambda_{1,2} = \pm \omega_O \sqrt{-3\sigma_{yy}} \quad (8.206)$$

which are either purely imaginary or one positive real and one negative real. Thus, the gravity-gradient pitch dynamics are marginally stable if $\sigma_{yy} > 0$ or $I_{xx} > I_{zz}$, otherwise it is unstable.

Furthermore, recalling that $I_{yy} < I_{xx} + I_{zz}$ must also hold and multiplying by $I_{xx} - I_{zz} > 0$ for pitch dynamics stability, one has

$$I_{yy}I_{xx} - I_{yy}I_{zz} < I_{xx}^2 - I_{zz}^2 \quad (8.207)$$

$$I_{xx}I_{yy} - I_{xx}^2 < I_{yy}I_{zz} - I_{zz}^2 \quad (8.208)$$

$$I_{xy}(I_{yy} - I_{xx}) < I_{zz}(I_{yy} - I_{zz}) \quad (8.209)$$

$$\frac{I_{yy} - I_{zz}}{I_{xx}} < \frac{I_{yy} - I_{zz}}{I_{xx}} \quad (8.210)$$

or, by definition, one has the alternative criterion for pitch dynamics stability

$$\sigma_{zz} < \sigma_{xx} \quad (8.211)$$

Ignoring the disturbances, the linearized roll-yaw state-space dynamics can be rearranged as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\psi} \\ \ddot{\phi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -4\omega_O^2\sigma_{xx} & 0 & 0 & \omega_O(1 - \sigma_{xx}) \\ 0 & -\omega_O^2\sigma_{zz} & -\omega_O(1 - \sigma_{zz}) & 0 \end{bmatrix} \begin{bmatrix} \phi \\ \psi \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ I_{xx}^{-1} & 0 \\ 0 & I_{zz}^{-1} \end{bmatrix} \begin{bmatrix} M_{c,x} \\ M_{c,z} \end{bmatrix} \quad (8.212)$$

with characteristic equation

$$\lambda^4 + (3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)\omega_O^2\lambda^2 + 4\sigma_{xx}\sigma_{zz}\omega_O^4 = 0 \quad (8.213)$$

and four roots

$$\lambda_{1,2,3,4} = \pm \omega_O \sqrt{\frac{1}{2} \left(-(3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1) \pm \sqrt{(3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 - 16\sigma_{xx}\sigma_{zz}} \right)} \quad (8.214)$$

with $\lambda_1 = -\lambda_2$ and $\lambda_3 = -\lambda_4$.

Thus, for all roots to not have any positive real parts, then all λ 's must be purely imaginary. This requires

$$-(3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1) \pm \sqrt{(3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 - 16\sigma_{xx}\sigma_{zz}} \quad (8.215)$$

be a purely real negative number for both terms. This requires the following three criteria

$$\begin{aligned} -(3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1) &< 0 \\ (3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 - 16\sigma_{xx}\sigma_{zz} &> 0 \\ -(3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1) &< \sqrt{(3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 - 16\sigma_{xx}\sigma_{zz}} \end{aligned} \quad (8.216)$$

$$\begin{aligned} 3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1 &> 0 \\ (3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 &> 16\sigma_{xx}\sigma_{zz} \end{aligned} \quad (8.217)$$

$$\begin{aligned} (3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 &> (3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 - 16\sigma_{xx}\sigma_{zz} \\ 3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1 &> 0 \\ 3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1 &> 4\sqrt{\sigma_{xx}\sigma_{zz}} \end{aligned} \quad (8.218)$$

$$\begin{aligned} (3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 &> (3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1)^2 - 16\sigma_{xx}\sigma_{zz} \\ 3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1 &> 0 \\ 3\sigma_{xx} + \sigma_{xx}\sigma_{zz} + 1 &> 4\sqrt{\sigma_{xx}\sigma_{zz}} \\ \sigma_{xx}\sigma_{zz} &> 0 \end{aligned} \quad (8.219)$$

By inspection, the third criterion demonstrates that satisfying the second criterion naturally satisfies the first criterion. Furthermore, squaring the second criterion on both sides, one has

$$9\sigma_{xx}^2 + \sigma_{xx}^2\sigma_{zz}^2 + 1 + 6\sigma_{xx}^2\sigma_{zz} + 2\sigma_{xx}\sigma_{zz} + 6\sigma_{xx} > 16\sigma_{xx}\sigma_{zz} \quad (8.220)$$

Thus, the gravity-gradient roll-yaw dynamics are marginally stable if and only if

$$\begin{aligned} (\sigma_{zz}^2 + 6\sigma_{zz} + 9)\sigma_{xx}^2 + (-14\sigma_{zz} + 6)\sigma_{xx} + 1 &> 0 \\ \sigma_{xx}\sigma_{zz} &> 0 \end{aligned} \quad (8.221)$$

Finally, recalling the pitch dynamics stability criterion, $\sigma_{zz} < \sigma_{xx}$, one can define two **gravity-gradient stability regions** for marginal stability of the gravity-gradient satellite dynamics. The first is

$$\sigma_{xx} > 0, \quad \sigma_{zz} > 0, \quad \sigma_{zz} < \sigma_{xx} \quad (8.222)$$

and the second can be stated by

$$\sigma_{xx} < 0, \quad \sigma_{zz} < 0, \quad \sigma_{zz} < \sigma_{xx}, \quad (\sigma_{zz}^2 + 6\sigma_{zz} + 9)\sigma_{xx}^2 + (-14\sigma_{zz} + 6)\sigma_{xx} + 1 > 0 \quad (8.223)$$

These two stability regions are plotted in the following figure. Stability region 2 is notably a very small region and is seldom used owing to practical structural difficulties. It is also important to note that as the linearized dynamics are only marginally stable, passive and/or active damping is typically necessary for gravity-gradient-stabilized satellites.

By definition of the moment of inertia ratios, stability region 1 equivalently requires

$$I_{yy} - I_{zz} > 0, \quad I_{yy} - I_{xx} > 0, \quad I_{xx} > I_{zz} \quad (8.224)$$

and incorporating the requirements due to the definition of the moments of inertia, one has

$$I_{xx} + I_{zz} > I_{yy} > I_{xx} > I_{zz} \quad (8.225)$$

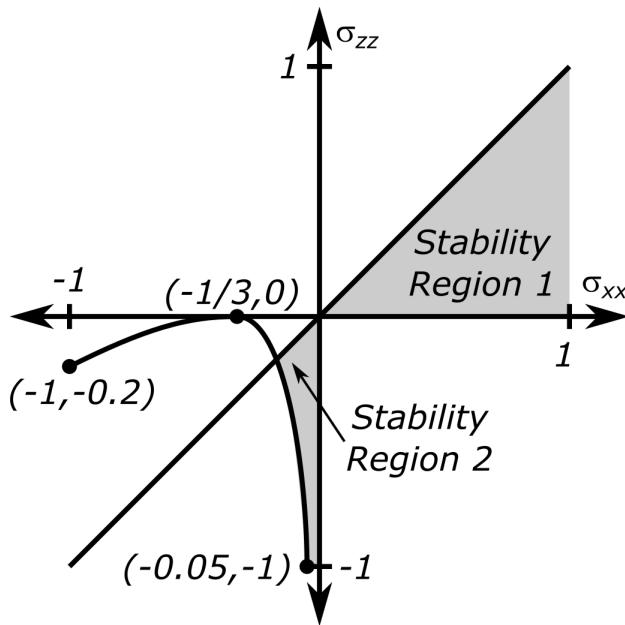
which imposes strict structural designs on a gravity-gradient-stabilized satellite in region 1. By definition of the moment of inertia ratios, stability region 2 equivalently requires

$$I_{yy} - I_{zz} < 0, \quad I_{yy} - I_{xx} < 0, \quad I_{xx} > I_{zz} \quad (8.226)$$

and incorporating the requirements due to the definition of the moments of inertia, one has

$$I_{yy} + I_{zz} > I_{xx} > I_{zz} > I_{yy}, \quad I_{yy} - I_{zz} \ll I_{xx} \quad (8.227)$$

which imposes even stricter criteria on a gravity-gradient-stabilized satellite in region 2 than region 1 which is why region 1 is preferred.



References

For more information, please refer to the following

- Curtis, H. D., “10.10 Gravity-Gradient Stabilization,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 530-540
- Sidi, M. J., “4.8 Attitude Dynamics Equations of Motion for a Nonspinning Spacecraft,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 107-110
- Sidi, M. J., “5.3 Gravity Gradient Attitude Control,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 114-129

Advanced Rigid Aerospace Vehicle Dynamics

9.1 Advanced Airplane Trim Analysis and Linearized Dynamics

Turning Steady-Flight Equations

First, recall the rigid-body **steady-flight conditions**, by definition, occur when the state variables in the rigid airplane EOMs are constant, i.e.

$$\dot{u} = \dot{\alpha} = \dot{\beta} = \dot{p} = \dot{q} = \dot{r} = \dot{\phi} = \dot{\theta} = 0 \quad (9.1)$$

which imply that the steady-flight conditions solve the **rigid-body steady-flight equations** in the stability frame, i.e., $\bar{\alpha} = 0$, as

$$\begin{bmatrix} \bar{X} - g \sin \bar{\gamma} \\ \bar{Y} + g \sin \bar{\phi} \cos \bar{\gamma} \\ \bar{Z} - g \cos \bar{\phi} \cos \bar{\gamma} \\ \bar{L}_{roll} \\ \bar{M} \\ \bar{N} \end{bmatrix} = \begin{bmatrix} -\bar{r}\bar{u} \tan \bar{\beta} \\ \bar{r}\bar{u} \\ \bar{p}\bar{u} \tan \bar{\beta} - \bar{q}\bar{u} \\ \frac{I_{zz}-I_{yy}}{I_{xx}} \bar{q}\bar{r} - \frac{I_{xz}}{I_{xx}} \bar{p}\bar{q} \\ \frac{I_{xx}-I_{zz}}{I_{yy}} \bar{p}\bar{r} + \frac{I_{xz}}{I_{yy}} (\bar{p}^2 - \bar{r}^2) \\ \frac{I_{yy}-I_{xz}}{I_{zz}} \bar{p}\bar{q} + \frac{I_{xz}}{I_{zz}} \bar{q}\bar{r} \end{bmatrix} \quad (9.2)$$

with the supplemental kinematic equations given as

$$\begin{bmatrix} \bar{p} \\ \bar{q} \\ \bar{r} \end{bmatrix} = \begin{bmatrix} -\dot{\psi} \sin \bar{\theta} \\ \dot{\psi} \sin \bar{\phi} \cos \bar{\theta} \\ \dot{\psi} \cos \bar{\phi} \cos \bar{\theta} \end{bmatrix} \quad (9.3)$$

where $\dot{\psi}$ is a constant. For $\dot{\psi} = 0$, one has straight steady-flight. However, for $\dot{\psi} \neq 0$, one has turning steady-flight.

For a simplified analysis, assume coordinated **level** turns, i.e. $\bar{\gamma} = 0$, the supplemental kinematic equations as

$$\begin{bmatrix} \bar{p} \\ \bar{q} \\ \bar{r} \end{bmatrix} = \begin{bmatrix} 0 \\ \dot{\psi} \sin \bar{\phi} \\ \dot{\psi} \cos \bar{\phi} \end{bmatrix} \quad (9.4)$$

where notably the steady-state pitch rate, $\bar{q} \neq 0$, which is due to the non-zero roll angle of the airplane required to turn. Furthermore, substituting for the aerodynamic and propulsive forces and moments, one has

$$\begin{bmatrix} -\bar{D} \cos \bar{\beta} - \bar{S} \sin \bar{\beta} + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ \bar{S} \cos \bar{\beta} - \bar{D} \sin \bar{\beta} + mg \sin \bar{\phi} \\ -\bar{L} - \bar{T} \sin(\theta_T + \bar{\alpha}) + mg \cos \bar{\phi} \\ \bar{L}_{a,S} \\ \bar{M}_{a,S} + \bar{T}(z_T \cos \theta_T - x_T \sin \theta_T) \\ \bar{N}_{a,S} \end{bmatrix} = \begin{bmatrix} -m\bar{r}\bar{u} \tan \bar{\beta} \\ m\bar{r}\bar{u} \\ -m\bar{q}\bar{u} \\ (I_{zz} - I_{yy}) \bar{q}\bar{r} \\ -I_{xz}\bar{r}^2 \\ I_{xz}\bar{q}\bar{r} \end{bmatrix} \quad (9.5)$$

which assumes the thrust force is symmetric with respect to the $x_B - z_B$ plane of the vehicle.

Next, assume that the lateral aerodynamic force is zero, i.e.

$$\bar{S} \cos \bar{\beta} - \bar{D} \sin \bar{\beta} = 0 \quad (9.6)$$

Then, the second and third equations for the lateral and vertical forces state

$$\begin{bmatrix} mg \sin \bar{\phi} \\ \bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) - mg \cos \bar{\phi} \end{bmatrix} = \begin{bmatrix} m\bar{r}\bar{u} \\ m\bar{q}\bar{u} \end{bmatrix} \quad (9.7)$$

and substituting in terms of $\dot{\psi}$, one has

$$\begin{bmatrix} mg \sin \bar{\phi} \\ \bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) - mg \cos \bar{\phi} \end{bmatrix} = \begin{bmatrix} m\bar{u}(\dot{\psi} \cos \bar{\phi}) \\ m\bar{u}(\dot{\psi} \sin \bar{\phi}) \end{bmatrix} \quad (9.8)$$

The first equation can be rewritten as

$$mg \tan \bar{\phi} = m\bar{u}\dot{\psi} \quad (9.9)$$

where one can note

$$\dot{\psi} = \frac{g}{\bar{u}} \tan \bar{\phi} \quad (9.10)$$

and substituting this into the second equation, one has

$$\bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) = mg (\cos \bar{\phi} + \tan \bar{\phi} \sin \bar{\phi}) \quad (9.11)$$

and multiplying by $\cos \bar{\phi}$, one has

$$\bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) \cos \bar{\phi} = mg (\cos^2 \bar{\phi} + \sin^2 \bar{\phi}) \quad (9.12)$$

$$\bar{L} + \bar{T} \sin(\theta_T + \bar{\alpha}) \cos \bar{\phi} = mg \quad (9.13)$$

With this in mind, one can define the dimensionless **normal load factor**, n , as

$$n(mg) = L + T \sin(\theta_T + \alpha) \quad (9.14)$$

which is typically referred to in terms of “g’s.” Thus, one has

$$\bar{n}(mg) = \frac{1}{\cos \bar{\phi}} \quad (9.15)$$

which is often specified instead of $\bar{\phi}$ for steady turns. However, one does require the direction of the turn to be specified in this case. Note that in wings-level flight, i.e., $\bar{\phi} = 0$, $\bar{n} = 1$ g.

Lastly, this can also be used to denote the pitch and yaw rates as

$$\begin{bmatrix} \bar{q} \\ \bar{r} \end{bmatrix} = \begin{bmatrix} \frac{g}{\bar{u}} \tan \bar{\phi} \sin \bar{\phi} \\ \frac{g}{\bar{u}} \sin \bar{\phi} \end{bmatrix} \quad (9.16)$$

or

$$\begin{bmatrix} \bar{q} \\ \bar{r} \end{bmatrix} = \begin{bmatrix} \frac{g}{\bar{u}} \left(\bar{n} - \frac{1}{\bar{n}} \right) \\ \pm \frac{g}{\bar{u}\bar{n}} \sqrt{\bar{n}^2 - 1} \end{bmatrix} \quad (9.17)$$

Level Turn Lateral-Directional Trim Analysis

With these derivations in mind which notably assume

$$\bar{S} \cos \bar{\beta} - \bar{D} \sin \bar{\beta} = 0 \quad (9.18)$$

and expressing the lateral-directional aerodynamic forces and moments in terms of the coefficients, C_S , C_l , and C_n , one has the following lateral-directional trim conditions for level turns

$$\begin{bmatrix} Q_\infty S_w \bar{C}_S \cos \bar{\beta} - Q_\infty S_w \bar{C}_D \sin \bar{\beta} \\ Q_\infty S_w \bar{C}_l \\ Q_\infty S_w \bar{C}_n \end{bmatrix} = \begin{bmatrix} 0 \\ (I_{zz} - I_{yy}) \bar{q} \bar{r} \\ I_{xz} \bar{q} \bar{r} \end{bmatrix} \quad (9.19)$$

and substituting for \bar{r} and \bar{q} , one has

$$\begin{bmatrix} Q_\infty S_w \bar{C}_S \cos \bar{\beta} - Q_\infty S_w \bar{C}_D \sin \bar{\beta} \\ Q_\infty S_w \bar{C}_l \\ Q_\infty S_w \bar{C}_n \end{bmatrix} = \begin{bmatrix} 0 \\ (I_{zz} - I_{yy}) \left(\frac{g}{\bar{u}} \right)^2 \tan \bar{\phi} \sin^2 \bar{\phi} \\ I_{xz} \left(\frac{g}{\bar{u}} \right)^2 \tan \bar{\phi} \sin^2 \bar{\phi} \end{bmatrix} \quad (9.20)$$

Furthermore, for a simplified analysis of these equations, one can assume linear relationships for these aerodynamic coefficients with respect to the trim drag coefficient, \bar{C}_D , trim side coefficient, \bar{C}_S , trim sideslip angle, $\bar{\beta}$, trim yaw rate, \bar{r} , trim aileron deflection $\bar{\delta}_a$, and trim rudder deflection, $\bar{\delta}_r$, i.e.

$$\bar{C}_S = C_{S_\beta} \bar{\beta} + C_{S_r} \bar{r} + C_{S_{\delta_a}} \bar{\delta}_a + C_{S_{\delta_r}} \bar{\delta}_r \quad (9.21)$$

$$\bar{C}_l = C_{l_\beta} \bar{\beta} + C_{l_r} \bar{r} + C_{l_{\delta_a}} \bar{\delta}_a + C_{l_{\delta_r}} \bar{\delta}_r \quad (9.22)$$

and

$$\bar{C}_n = C_{n\beta} \bar{\beta} + C_{n_r} \bar{r} + C_{n_{\delta_a}} \bar{\delta}_a + C_{n_{\delta_r}} \bar{\delta}_r \quad (9.23)$$

Then, by substitution for the coefficients and \bar{r} in terms of $\bar{\phi}$, one has

$$\begin{bmatrix} Q_\infty S_w \left(C_{S\beta} \bar{\beta} + C_{S_r} \frac{g}{\bar{u}} \sin \bar{\phi} + C_{S_{\delta_a}} \bar{\delta}_a + C_{S_{\delta_r}} \bar{\delta}_r \right) \cos \bar{\beta} - Q_\infty S_w \bar{C}_D \sin \bar{\beta} \\ Q_\infty S_w b_w \left(C_{l\beta} \bar{\beta} + C_{l_r} \frac{g}{\bar{u}} \sin \bar{\phi} + C_{l_{\delta_a}} \bar{\delta}_a + C_{l_{\delta_r}} \bar{\delta}_r \right) \\ Q_\infty S_w b_w \left(C_{n\beta} \bar{\beta} + C_{n_r} \frac{g}{\bar{u}} \sin \bar{\phi} + C_{n_{\delta_a}} \bar{\delta}_a + C_{n_{\delta_r}} \bar{\delta}_r \right) \end{bmatrix} = \begin{bmatrix} 0 \\ (I_{zz} - I_{yy}) \left(\frac{g}{\bar{u}} \right)^2 \tan \bar{\phi} \sin^2 \bar{\phi} \\ I_{xz} \left(\frac{g}{\bar{u}} \right)^2 \tan \bar{\phi} \sin^2 \bar{\phi} \end{bmatrix} \quad (9.24)$$

Thus, for a given Q_∞ , i.e. a given airspeed \bar{u} and altitude, and roll angle, $\bar{\phi}$ or normal load factor \bar{n} , these equations determine the three unknowns for the lateral-directional trim conditions, i.e. $\bar{\beta}$, $\bar{\delta}_a$, and $\bar{\delta}_r$. This can be solved using numerical methods. However, it does require that \bar{C}_D has already been obtained from the longitudinal trim analysis. However, the longitudinal analysis depends on the trim sideslip angle, $\bar{\beta}$, thus one typically must iterate between these trim computations with the initial attempt typically assuming $\bar{C}_D \approx 0$.

An analytical approach is to assume $\bar{\beta}$ is small, i.e. $\sin \bar{\beta} \approx \bar{\beta}$ and $\cos \bar{\beta} \approx 1$, then, one has

$$\begin{bmatrix} \bar{C}_{S\beta} - \bar{C}_D & C_{S_{\delta_a}} & C_{S_{\delta_r}} \\ C_{l\beta} & C_{l_{\delta_a}} & C_{l_{\delta_r}} \\ C_{n\beta} & C_{n_{\delta_a}} & C_{n_{\delta_r}} \end{bmatrix} \begin{bmatrix} \bar{\beta} \\ \bar{\delta}_a \\ \bar{\delta}_r \end{bmatrix} = \begin{bmatrix} -C_{S_r} \frac{g}{\bar{u}} \sin \bar{\phi} \\ \left(\frac{I_{zz} - I_{yy}}{Q_\infty S_w b_w} \right) \left(\frac{g}{\bar{u}} \right)^2 \tan \bar{\phi} \sin^2 \bar{\phi} - C_{l_r} \frac{g}{\bar{u}} \sin \bar{\phi} \\ \frac{I_{xz}}{Q_\infty S_w b_w} \left(\frac{g}{\bar{u}} \right)^2 \tan \bar{\phi} \sin^2 \bar{\phi} - C_{n_r} \frac{g}{\bar{u}} \sin \bar{\phi} \end{bmatrix} \quad (9.25)$$

or, in terms of the trim normal load factor, one has

$$\begin{bmatrix} \bar{C}_{S\beta} - \bar{C}_D & C_{S_{\delta_a}} & C_{S_{\delta_r}} \\ C_{l\beta} & C_{l_{\delta_a}} & C_{l_{\delta_r}} \\ C_{n\beta} & C_{n_{\delta_a}} & C_{n_{\delta_r}} \end{bmatrix} \begin{bmatrix} \bar{\beta} \\ \bar{\delta}_a \\ \bar{\delta}_r \end{bmatrix} = \begin{bmatrix} \mp C_{S_r} \pm \frac{g}{\bar{u}\bar{n}} \sqrt{\bar{n}^2 - 1} \\ \pm \left(\frac{I_{zz} - I_{yy}}{Q_\infty S_w b_w} \right) \left(\frac{g}{\bar{u}} \right)^2 \left(1 - \frac{1}{\bar{n}^2} \right) \sqrt{\bar{n}^2 - 1} \mp C_{l_r} \frac{g}{\bar{u}\bar{n}} \sqrt{\bar{n}^2 - 1} \\ \pm \frac{I_{xz}}{Q_\infty S_w b_w} \left(\frac{g}{\bar{u}} \right)^2 \left(1 - \frac{1}{\bar{n}^2} \right) \sqrt{\bar{n}^2 - 1} \mp C_{n_r} \frac{g}{\bar{u}\bar{n}} \sqrt{\bar{n}^2 - 1} \end{bmatrix} \quad (9.26)$$

which either can be solved by multiplying both sides by the inverse matrix on the left side.

Level Turn Longitudinal Trim Analysis

Thus, having solved for the trim sideslip angle, $\bar{\beta}$, trim side force coefficient, \bar{C}_S , trim aileron deflection, $\bar{\delta}_a$, and trim rudder deflection $\bar{\delta}_r$, one has the following longitudinal trim conditions for level turns

$$\begin{bmatrix} -\bar{D} \cos \bar{\beta} - \bar{S} \sin \bar{\beta} + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ -\bar{L} - \bar{T} \sin(\theta_T + \bar{\alpha}) + mg \cos \bar{\phi} \\ \bar{M}_a + \bar{T}(z_T \cos \theta_T - x_T \sin \theta_T) \end{bmatrix} = \begin{bmatrix} -m\bar{r}\bar{u} \tan \bar{\beta} \\ -m\bar{q}\bar{u} \\ -I_{xz}\bar{r}^2 \end{bmatrix} \quad (9.27)$$

Next, one can express the longitudinal aerodynamic forces and moments in terms of the coefficients, C_L , C_S , C_D , and C_m , as

$$\begin{bmatrix} -Q_\infty S_w \bar{C}_D \cos \bar{\beta} - Q_\infty S_w \bar{C}_S \sin \bar{\beta} + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ Q_\infty S_w \bar{C}_L + \bar{T} \sin(\theta_T + \bar{\alpha}) \\ Q_\infty S_w \bar{C}_m + \bar{T}(z_T \cos \theta_T - x_T \sin \theta_T) \end{bmatrix} = \begin{bmatrix} -m\bar{r}\bar{u} \tan \bar{\beta} \\ -m\bar{q}\bar{u} \\ -I_{xz}\bar{r}^2 \end{bmatrix} \quad (9.28)$$

Furthermore, for a simplified analysis of these equations, one can assume linear relationships for C_L , C_D , and C_m with respect to the trim angle of attack, $\bar{\alpha}$, trim pitch rate, \bar{q} , trim elevator deflection, $\bar{\delta}_e$, trim aileron deflection, $\bar{\delta}_a$, and trim rudder deflection $\bar{\delta}_r$, i.e.,

$$\bar{C}_D = C_{D_0} + C_{D_\alpha} \bar{\alpha} + C_{D_q} \bar{q} + C_{D_{\delta_e}} \bar{\delta}_e + C_{D_{\delta_a}} \bar{\delta}_a + C_{D_{\delta_r}} \bar{\delta}_r \quad (9.29)$$

$$\bar{C}_L = C_{L_0} + C_{L_\alpha} \bar{\alpha} + C_{L_q} \bar{q} + C_{L_{\delta_e}} \bar{\delta}_e \quad (9.30)$$

and

$$\bar{C}_m = C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_q} \bar{q} + C_{m_{\delta_e}} \bar{\delta}_e \quad (9.31)$$

Then, by substitution, one has

$$\begin{bmatrix} -Q_\infty S_w \left(C_{D_0} + C_{D_\alpha} \bar{\alpha} + C_{D_q} \bar{q} + C_{D_{\delta_e}} \bar{\delta}_e + C_{D_{\delta_a}} \bar{\delta}_a + C_{D_{\delta_r}} \bar{\delta}_r \right) \cos \bar{\beta} - Q_\infty S_w \bar{C}_S \sin \bar{\beta} + \bar{T} \cos(\theta_T + \bar{\alpha}) \\ Q_\infty S_w \left(C_{L_0} + C_{L_\alpha} \bar{\alpha} + C_{L_q} \bar{q} + C_{L_{\delta_e}} \bar{\delta}_e \right) + \bar{T} \sin(\theta_T + \bar{\alpha}) \\ Q_\infty S_w \bar{C}_w \left(C_{m_0} + C_{m_\alpha} \bar{\alpha} + C_{m_q} \bar{q} + C_{m_{\delta_e}} \bar{\delta}_e \right) + \bar{T} \frac{z_T \cos \theta_T - x_T \sin \theta_T}{Q_\infty S_w \bar{C}_w} \end{bmatrix} = \begin{bmatrix} -m \bar{r} \bar{u} \tan \bar{\beta} \\ -m \bar{q} \bar{u} \\ -I_{xz} \bar{r}^2 \end{bmatrix} \quad (9.32)$$

Thus, for a given Q_∞ , i.e. a given airspeed \bar{u} and altitude, and roll angle, $\bar{\phi}$ or normal load factor \bar{n} , these equations determine the three unknowns for the longitudinal trim conditions, i.e. $\bar{\alpha}$, $\bar{\delta}_e$, and \bar{T} . This can be solved using numerical methods.

An analytical approach is to assume $\bar{\beta}$ is small, i.e. $\sin \bar{\beta} \approx \bar{\beta}$ and $\cos \bar{\beta} \approx 1$, $\cos(\theta_T + \bar{\alpha}) = 1$, and $\sin(\theta_T + \bar{\alpha}) = 0$, then, one has

$$\begin{bmatrix} -C_{D_\alpha} & -C_{D_{\delta_e}} & \frac{1}{Q_\infty S_w} \\ C_{L_\alpha} & C_{L_{\delta_e}} & 0 \\ C_{m_\alpha} & C_{m_{\delta_e}} & \frac{z_T \cos \theta_T - x_T \sin \theta_T}{Q_\infty S_w \bar{C}_w} \end{bmatrix} \begin{bmatrix} \bar{\alpha} \\ \bar{\delta}_e \\ \bar{T} \end{bmatrix} = \begin{bmatrix} \left(\bar{C}_S - \frac{mg}{Q_\infty S_w} \sin \bar{\phi} \right) \bar{\beta} + C_{D_0} + C_{D_q} \frac{g}{\bar{u}} \tan \bar{\phi} \sin \bar{\phi} + C_{D_{\delta_a}} \bar{\delta}_a + C_{D_{\delta_r}} \bar{\delta}_r \\ \frac{mg}{Q_\infty S_w \cos \bar{\phi}} - C_{L_0} - C_{L_q} \frac{g}{\bar{u}} \tan \bar{\phi} \sin \bar{\phi} \\ -I_{xz} \left(\frac{g}{\bar{u}} \right)^2 \sin^2 \bar{\phi} - C_{m_0} - C_{m_q} \frac{g}{\bar{u}} \tan \bar{\phi} \sin \bar{\phi} \end{bmatrix} \quad (9.33)$$

or, in terms of the trim normal load factor, one has

$$\begin{bmatrix} -C_{D_\alpha} & -C_{D_{\delta_e}} & \frac{1}{Q_\infty S_w} \\ C_{L_\alpha} & C_{L_{\delta_e}} & 0 \\ C_{m_\alpha} & C_{m_{\delta_e}} & \frac{z_T \cos \theta_T - x_T \sin \theta_T}{Q_\infty S_w \bar{C}_w} \end{bmatrix} \begin{bmatrix} \bar{\alpha} \\ \bar{\delta}_e \\ \bar{T} \end{bmatrix} = \begin{bmatrix} \left(\bar{C}_S - \frac{mg}{Q_\infty S_w} \sqrt{1 - \frac{1}{\bar{n}^2}} \right) \bar{\beta} + C_{D_0} + C_{D_q} \frac{g}{\bar{u}} \tan \bar{\phi} \sin \bar{\phi} + C_{D_{\delta_a}} \bar{\delta}_a + C_{D_{\delta_r}} \bar{\delta}_r \\ \frac{mg}{Q_\infty S_w} - C_{L_0} - C_{L_q} \frac{g}{\bar{u}} \left(\bar{n} - \frac{1}{\bar{n}} \right) \\ -I_{xz} \left(\frac{g}{\bar{u}} \right)^2 \left(1 - \frac{1}{\bar{n}^2} \right) - C_{m_0} - C_{m_q} \frac{g}{\bar{u}} \left(\bar{n} - \frac{1}{\bar{n}} \right) \end{bmatrix} \quad (9.34)$$

which either can be solved by multiplying both sides by the inverse matrix on the left side.

However, if one alternatively assumes $z_T \cos \theta_T - x_T \sin \theta_T \approx 0$, then one has a second method via

$$\begin{bmatrix} -C_{D_\alpha} & -C_{D_{\delta_e}} & \frac{\cos(\theta_T + \bar{\alpha})}{Q_\infty S_w} \\ C_{L_\alpha} & C_{L_{\delta_e}} & 0 \\ C_{m_\alpha} & C_{m_{\delta_e}} & 0 \end{bmatrix} \begin{bmatrix} \bar{\alpha} \\ \bar{\delta}_e \\ \bar{T} \end{bmatrix} = \begin{bmatrix} \left(\bar{C}_S - \frac{mg}{Q_\infty S_w} \sqrt{1 - \frac{1}{\bar{n}^2}} \right) \bar{\beta} + C_{D_0} + C_{D_q} \frac{g}{\bar{u}} \tan \bar{\phi} \sin \bar{\phi} + C_{D_{\delta_a}} \bar{\delta}_a + C_{D_{\delta_r}} \bar{\delta}_r \\ \bar{n} \frac{mg}{Q_\infty S_w} - C_{L_0} - C_{L_q} \frac{g}{\bar{u}} \left(\bar{n} - \frac{1}{\bar{n}} \right) \\ -I_{xz} \left(\frac{g}{\bar{u}} \right)^2 \left(1 - \frac{1}{\bar{n}^2} \right) - C_{m_0} - C_{m_q} \frac{g}{\bar{u}} \left(\bar{n} - \frac{1}{\bar{n}} \right) \end{bmatrix} \quad (9.35)$$

which allows one to decouple $\bar{\alpha}$ and $\bar{\delta}_e$ from \bar{T} . This results in the analytical solution

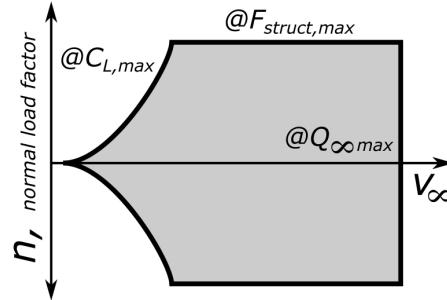
$$\begin{aligned}\bar{\alpha} &\approx \frac{C_1(n)C_{m_{\delta_e}} - C_2(n)C_{L_{\delta_e}}}{C_{L_\alpha}C_{m_{\delta_e}} - C_{m_\alpha}C_{L_{\delta_e}}} \\ \bar{\delta}_e &\approx -\frac{C_1(n)C_{m_\alpha} - C_2(n)C_{L_\alpha}}{C_{L_\alpha}C_{m_{\delta_e}} - C_{m_\alpha}C_{L_{\delta_e}}} \\ C_1(n) &= \bar{n} \frac{mg}{Q_\infty S_w} - C_{L_0} - C_{L_q} \frac{g}{\bar{u}} \left(\bar{n} - \frac{1}{\bar{n}} \right) \\ C_2(n) &= -I_{xz} \left(\frac{g}{\bar{u}} \right)^2 \left(1 - \frac{1}{\bar{n}^2} \right) - C_{m_0} - C_{m_q} \frac{g}{\bar{u}} \left(\bar{n} - \frac{1}{\bar{n}} \right)\end{aligned}\quad (9.36)$$

which is only a function of the normal load factor and the lift and M -moment coefficients. Then, one can determine the trim thrust via the drag equation

$$\bar{T} = \frac{-mg \sqrt{1 - \frac{1}{\bar{n}^2}} \tan \bar{\beta} + Q_\infty S_w \bar{C}_D \cos \bar{\beta} + Q_\infty S_w \bar{C}_S \sin \bar{\beta}}{\cos(\theta_T + \bar{\alpha})} \quad (9.37)$$

which notably is a function of the drag coefficients and the trim angle of attack and elevator deflection.

As for the flight envelope, the lift coefficient, $C_{L,max}$, dynamic pressure, $Q_{\infty,max}$, and structural limits, $F_{struct,max}$, for level, turning flight define the vehicle's **v-n plot** plotting airspeed, v_∞ , versus the normal load factor, n , as



where $C_{L,max}$ results from either the maximum angle of attack or the maximum elevator deflection, $Q_{\infty,max}$ is often due to a flutter limit, and the center region defines the possible steady-flight conditions for the airplane.

Coupled Airplane Linearized Dynamics

For non-coordinated, non-wings-level steady-flight conditions, including turning flight, the linearized longitudinal and lateral-directional state-space models cannot be decoupled, but become coupled. Often this coupling is weak and can be ignored in the control design. However, under certain trim conditions, e.g., high angles of attack or sideslip, the fully coupled equations are necessary in order to assess the coupling effects of the control inputs on all the airplane states. This coupling also typically uses coupled stability and control

derivatives, e.g. X_β , Y_α , L_α^* , N_α^* , and X_{δ_r} . This section presents the coupled airplane linearized dynamics using the **polynomial-matrix model** for more clarity in the expressions, i.e.,

$$P(s)\vec{y}(s) = Q(s)\vec{u}(s) \quad (9.38)$$

where the system transfer function matrix is given by

$$[G(s)] = P^{-1}(s)Q(s) \quad (9.39)$$

For the coupled polynomial-matrix model, one has

$$\begin{bmatrix} P_{long}(s) & P_{long-lat}(s) \\ P_{lat-long}(s) & P_{lat}(s) \end{bmatrix} \begin{bmatrix} \vec{y}_{long}(s) \\ \vec{y}_{lat}(s) \end{bmatrix} = \begin{bmatrix} Q_{long}(s) & Q_{long-lat}(s) \\ Q_{lat-long}(s) & Q_{lat}(s) \end{bmatrix} \begin{bmatrix} \vec{u}_{long}(s) \\ \vec{u}_{lat}(s) \end{bmatrix} \quad (9.40)$$

where the output vectors are

$$\vec{y}_{long}(s) = \begin{bmatrix} \Delta u(s) \\ \Delta \alpha(s) \\ \Delta q(s) \\ \Delta \theta(s) \end{bmatrix} \quad (9.41)$$

$$\vec{y}_{lat}(s) = \begin{bmatrix} \Delta \beta(s) \\ \Delta p(s) \\ \Delta r(s) \\ \Delta \phi(s) \\ \Delta \psi(s) \end{bmatrix} \quad (9.42)$$

the input vectors are

$$\vec{u}_{long}(s) = \begin{bmatrix} \Delta \delta_e(s) \\ \Delta \delta_T(s) \end{bmatrix} \quad (9.43)$$

$$\vec{u}_{lat}(s) = \begin{bmatrix} \Delta \delta_a(s) \\ \Delta \delta_r(s) \end{bmatrix} \quad (9.44)$$

the diagonal polynomial-matrices are

$$P_{long}(s) = \begin{bmatrix} s - X_u & -X_\alpha + \bar{q}\bar{u} & \bar{w} - X_q & g \cos \bar{\theta} \\ -Z_u - \bar{q} & (\bar{u} - Z_\alpha)s - Z_\alpha & -Z_q - \bar{u} & g \sin \bar{\theta} \cos \bar{\phi} \\ -M_u & -M_{\dot{\alpha}}s - M_\alpha & s - M_q & 0 \\ 0 & 0 & -\cos \bar{\phi} & s \end{bmatrix} \quad (9.45)$$

$$Q_{long}(s) = \begin{bmatrix} X_{\delta_e} & X_{\delta_T} \\ Z_{\delta_e} & Z_{\delta_T} \\ M_{\delta_e} & M_{\delta_T} \\ 0 & 0 \end{bmatrix} \quad (9.46)$$

$$P_{lat}(s) = \begin{bmatrix} \bar{u}s - Y_\beta & -Y_p - \bar{w} & \bar{u} - Y_r & -g \cos \bar{\theta} \cos \bar{\phi} & 0 \\ -L_\beta^* & s - L_p^* - C_1\bar{q} & -L_r^* - C_2\bar{q} & 0 & 0 \\ -N_\beta^* & -N_p^* - C_3\bar{q} & s - N_r^* + C_1\bar{q} & 0 & 0 \\ 0 & 1 & \tan \bar{\theta} \cos \bar{\phi} & -s + (\bar{q} \cos \bar{\phi} - \bar{r} \sin \bar{\phi}) \tan \bar{\theta} & 0 \\ 0 & 0 & \cos \bar{\phi} & \bar{q} \cos \bar{\phi} - \bar{r} \sin \bar{\phi} & -s \cos \bar{\phi} \end{bmatrix} \quad (9.47)$$

where

$$L_{\bullet}^* = \left(L_{\bullet} + N_{\bullet} \frac{I_{xz}}{I_{zz}} \right) D_{xz} \quad (9.48)$$

$$N_{\bullet}^* = \left(N_{\bullet} + L_{\bullet} \frac{I_{xz}}{I_{zz}} \right) D_{xz} \quad (9.49)$$

$$D_{xz} = \left(1 - \frac{I_{xz}^2}{I_{xx} I_{zz}} \right)^{-1} \quad (9.50)$$

for $\bullet = \beta, p, r, \delta_a$, and δ_r due to the coupling of L and N for non-zero I_{xz} , and

$$C_1 = (I_{xx} - I_{yy} + I_{zz}) I_{xz} I_{xx}^{-1} I_{zz}^{-1} D_{xz} \quad (9.51)$$

$$C_2 = \left(I_{yy} - I_{zz} + \frac{I_{xz}^2}{I_{zz}} \right) I_{xx}^{-1} D_{xz} \quad (9.52)$$

$$C_3 = \left(I_{xx} - I_{yy} + \frac{I_{xz}^2}{I_{xx}} \right) I_{zz}^{-1} D_{xz} \quad (9.53)$$

$$Q_{lat}(s) = \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a}^* & L_{\delta_r}^* \\ N_{\delta_a}^* & N_{\delta_r}^* \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (9.54)$$

and the off-diagonal polynomial-matrices are

$$P_{long-lat}(s) = \begin{bmatrix} -\bar{r}\bar{u} - X_{\beta} & 0 & -\bar{\beta}\bar{u} & 0 & 0 \\ \bar{p}\bar{u} & \bar{\beta}\bar{u} & 0 & g \cos \bar{\theta} \sin \bar{\phi} & 0 \\ 0 & (I_{xx} - I_{zz}) I_{yy}^{-1} \bar{r} + 2I_{xz} I_{yy}^{-1} \bar{p} & (I_{xx} - I_{zz}) I_{yy}^{-1} \bar{p} - 2I_{xz} I_{yy}^{-1} \bar{r} & 0 & 0 \\ 0 & 0 & \sin \bar{\phi} & \dot{\psi} \cos \bar{\theta} & 0 \end{bmatrix} \quad (9.55)$$

$$Q_{long-lat}(s) = \begin{bmatrix} 0 & X_{\delta_r} \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \quad (9.56)$$

$$P_{lat-long}(s) = \begin{bmatrix} \bar{r} & \bar{p}\bar{u} - Y_{\alpha} & 0 & g \sin \bar{\theta} \sin \bar{\phi} \\ 0 & -L_{\alpha}^* & I_{zz}^{-1} D_{xz} C_4 + I_{xz} I_{xx}^{-1} I_{zz}^{-1} D_{xz} C_5 & 0 \\ 0 & -N_{\alpha}^* & I_{xx}^{-1} D_{xz} C_5 + I_{xz} I_{xx}^{-1} I_{zz}^{-1} D_{xz} C_4 & 0 \\ 0 & 0 & \tan \bar{\theta} \sin \bar{\phi} & \bar{q} \sin \bar{\phi} + \bar{r} \cos \bar{\phi} + \dot{\psi} \sin \bar{\theta} \tan \bar{\theta} \\ 0 & 0 & \sin \bar{\phi} & \dot{\psi} \sin \bar{\theta} \end{bmatrix} \quad (9.57)$$

and

$$C_4 = (I_{zz} - I_{yy}) \bar{r} - I_{xz} \bar{p} \quad (9.58)$$

$$C_5 = (I_{yy} - I_{xx}) \bar{p} + I_{xz} \bar{r} \quad (9.59)$$

$$Q_{lat-long}(s) = 0_{5 \times 2} \quad (9.60)$$

9.2 Advanced Mass Effects on Aerospace Vehicle Dynamics

Gravity Effect and Models

For flight dynamics, a high fidelity gravity model can be obtained by expanding Newton's point-mass Law of Gravitation in the ECEF frame using the second-order term of the Earth's gravitation field, J_2 , which yields

$$\vec{G}_E = -\frac{GM_E}{\|\vec{x}_E\|_2^3} \begin{bmatrix} \left(1 + \frac{3}{2} \left(\frac{R_e}{\|\vec{x}_E\|_2}\right)^2 J_2 (1 - 5 \sin^2 \ell)\right) x_E \\ \left(1 + \frac{3}{2} \left(\frac{R_e}{\|\vec{x}_E\|_2}\right)^2 J_2 (1 - 5 \sin^2 \ell)\right) y_E \\ \left(1 + \frac{3}{2} \left(\frac{R_e}{\|\vec{x}_E\|_2}\right)^2 J_2 (3 - 5 \sin^2 \ell)\right) z_E \end{bmatrix} \quad (9.61)$$

where $\vec{x}_E = [x_E \ y_E \ z_E]^T$ is the position of the flight vehicle's center of mass in the ECEF frame, R_e : Earth's equatorial radius, 6378137.0 m, Earth's gravitational parameter is

$$GM_E = 3986004.418 \times 10^8 \text{ m}^3/\text{s}^2 \quad (9.62)$$

and the second-order term by the **WGS 84 J_2 parameter**

$$J_2 = -\sqrt{5} \bar{C}_{2,0} = 0.001082626684 \quad (9.63)$$

where $\bar{C}_{2,0}$ is the actual specified parameter in the WGS 84 model. This is part of **Earth Gravitational Models (EGM)** published by the National Geospatial-Intelligence Agency (NGA).

This gravitational attraction minus the centripetal acceleration of the rotating Earth contributes to the weight force on the flight vehicle as

$$\vec{F}_{g,E} = m \vec{g}_E = m \left(\vec{G}_E - [\vec{\omega}_{E/I}]_\times [\vec{\omega}_{E/I}]_\times \vec{x}_E \right) \quad (9.64)$$

where \vec{g} is the **gravity vector** present in the flight vehicle EOMs. Often, one assumes this vector acts straight downward in the navigation frame, i.e.

$$\vec{F}_{g,N} = m \vec{g}_N \approx \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (9.65)$$

and will generally be a function of latitude ℓ and altitude h .

A lower-fidelity model for the **altitude correction of gravity** for the altitude *above MSL* is the **free air correction (FAC)**, as

$$g(h) = g_0 \left(\frac{\bar{R}_E}{\bar{R}_E + h} \right)^2 \quad (9.66)$$

where \bar{R}_E is Earth's *mean* radius, 6,371,000 m. As a linear approximation for $h \ll \bar{R}_E$, one has

$$g(h) \approx g_0 - \frac{2g_0}{\bar{R}_E^2} h = g_0 - 3.086 \times 10^{-6} h \quad (9.67)$$

However, as the Earth is truly a geoid with a variable gravitational field due to local features such as mountains, oceans, lakes, and even ore deposits, it is often useful to use a model for the gravitational force, $\vec{F}_g = m\vec{g}$, present in the flight vehicle EOMs with higher fidelity.

A lower-fidelity model for the **latitude correction of gravity** at MSL is the **WGS 84 Ellipsoidal Gravity Formula**, which is defined as

$$g_0(\ell) = g_e \left(\frac{1 + k \sin^2 \ell}{\sqrt{1 - e^2 \sin^2 \ell}} \right) \quad (9.68)$$

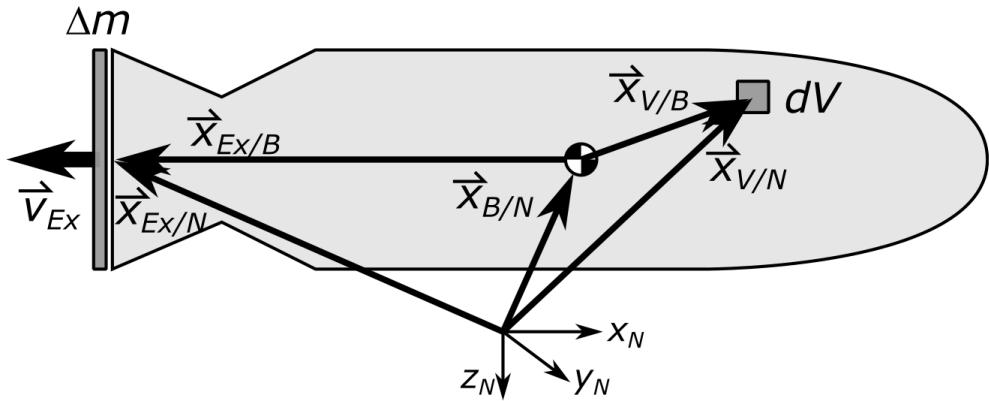
where $e^2 = 1 - (R_p/R_e)^2$ is the Earth's eccentricity squared, ≈ 0.00669437999014 , $k = \frac{R_p g_p - R_e g_e}{R_e g_e}$ is a formula constant, ≈ 0.001931853 , R_e is Earth's equatorial radius = 6378137.0 m, R_p is Earth's polar radius, 6356752.3 m, g_e is the acceleration due to gravity at the equator, ≈ 9.7803253359 m/s², and g_p is the acceleration due to gravity at the poles, 9.8321849378 m/s². Then, the **Somigliana gravity model** also provides the ellipsoidal height dependence as

$$g(\ell, h) = g_0(\ell) \left(1 - 2 \left(1 + f + \frac{a^3(1-f)\Omega_{I \leftarrow E}^2}{\mu_E} \right) \left(\frac{h}{a} \right) + 3 \left(\frac{h}{a} \right)^2 \right) \quad (9.69)$$

Mass Effect

Another mass effect is associated with the production of propulsive thrust through expelling mass. While this includes combustion jet engines that combust expelled fuel in air-breathing vehicles, the variable mass effects are typically only significant with rocket engines where the expelled mass includes the fuel *and* oxidizer for the combustion. Note that this derivation will consider the each of these position vectors as represented in inertial navigation frame N coordinates unless otherwise noted.

Thus, to this end, consider the position vectors of the various elements of a rocket as shown below



where the rocket's inertial *instantaneous* body frame center, i.e. the center of mass, is $\vec{x}_{B/N}$, the mass element of the vehicle, $\rho_V dV$, with position vectors relative to the body-fixed and navigation frames as $\vec{x}_{V/B}$ and $\vec{x}_{V/N}$, and the expelled mass Δm has position vectors relative to the body-fixed and navigation frames as $\vec{x}_{Ex/B}$ and $\vec{x}_{Ex/N}$ with \vec{v}_{Ex} as the **exit velocity** of Δm relative to the vehicle whose magnitude is often

specified for rocket engines by

$$\|\vec{v}_{Ex}\|_2^2 = I_{sp}g_0 \quad (9.70)$$

where I_{sp} is the specific impulse of the rocket engine and g_0 is the standard acceleration due to gravity.

At time t , one has for the linear momentum of the variable mass vehicle in the inertial navigation frame

$$\vec{P}_N(t) = \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} dV \quad (9.71)$$

while at time $t + \Delta t$, the linear momentum is

$$\vec{P}_N(t + \Delta t) = \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} + \Delta \dot{\vec{x}}_{V/N} \rho_V dV + \Delta m (\dot{\vec{x}}_{Ex/N} + \Delta \dot{\vec{x}}_{Ex/N}) \quad (9.72)$$

where the last term represents the linear momentum balance due to expelled mass, i.e. $\Delta m < 0$ for expelled mass. Then taking the limit as $\Delta t \rightarrow 0$, one has

$$\dot{\vec{P}}_N = \int_{Vol} \frac{d}{dt} (\rho_V \dot{\vec{x}}_{V/N}) dV + \dot{m} \dot{\vec{x}}_{Ex/N} \quad (9.73)$$

Comparing this expression with Newton's translation equation of motion, one can see that the total rate of change in the translational momentum may be rewritten as

$$\dot{\vec{P}}_N = \int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} \dot{\vec{x}}_{Ex/N} \quad (9.74)$$

where dF_{ext} is the external force acting at some infinitesimal surface area, i.e. pressure forces. Note that the inertial velocity of the expelled mass dm is

$$\dot{\vec{x}}_{Ex/N} = \vec{v}_{Ex} + \vec{v}_{B/N} + \vec{\omega}_{B/N} \times \vec{x}_{Ex/B} \quad (9.75)$$

Thus,

$$\dot{\vec{P}}_N = \int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} (\vec{v}_{Ex} + \vec{v}_{B/N} + \vec{\omega}_{B/N} \times \vec{x}_{Ex/B}) \quad (9.76)$$

Furthermore, from the definition of the center of mass at time t , one has

$$m \vec{x}_{B/N} = \int_{Vol} \rho_V \vec{x}_{V/N} dV \quad (9.77)$$

and at time $t + \Delta t$, one has

$$m \vec{x}_{B/N} + \Delta m \vec{x}_{B/N} = \int_{Vol} \rho_V (\vec{x}_{V/N} + \Delta \vec{x}_{V/N}) dV + \Delta m (\vec{x}_{Ex/N} + \Delta \vec{x}_{Ex/N}) \quad (9.78)$$

then as $\Delta t \rightarrow 0$, one has

$$\frac{d}{dt} (m \vec{x}_{B/N}) = \vec{P}_N(t) + \dot{m} \dot{\vec{x}}_{Ex/N} \quad (9.79)$$

where notably $\dot{m} < 0$. However, one can rewrite the position of the expelled mass dm as

$$\vec{x}_{Ex/N} = \vec{x}_{B/N} + \vec{x}_{Ex/B} \quad (9.80)$$

Thus, by the chain rule, one has

$$\dot{m} \vec{x}_{B/N} + m \vec{\dot{x}}_{B/N} = \vec{P}_N(t) + \dot{m} (\vec{x}_{B/N} + \vec{x}_{Ex/B}) \quad (9.81)$$

Therefore, the translational momentum can be rewritten as

$$\vec{P}_N(t) = m \vec{\dot{x}}_{B/N} - \dot{m} \vec{x}_{Ex/B} \quad (9.82)$$

and differentiating results in

$$\dot{\vec{P}}_N = m \ddot{\vec{x}}_{B/N} + \dot{m} (\vec{\dot{x}}_{B/N} - \vec{\dot{x}}_{Ex/B}) - \ddot{m} \vec{x}_{Ex/B} \quad (9.83)$$

where the inertial velocity of expelled mass dm is

$$\vec{\dot{x}}_{Ex/N} = \vec{v}_{B/N} + \vec{\omega}_{B/N} \times \vec{x}_{Ex/B} + \vec{v}_{Ex} \quad (9.84)$$

Combining Equations 9.76 and 9.83, one has

$$\begin{aligned} & m \ddot{\vec{x}}_{B/N} + \dot{m} (\vec{\dot{x}}_{B/N} - \vec{\dot{x}}_{Ex/B}) - \ddot{m} \vec{x}_{Ex/B} \\ &= \int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} (\vec{v}_{B/N} + \vec{x}_{Ex/B} + \vec{v}_{Ex}) \end{aligned} \quad (9.85)$$

Then, taking gravity to be constant over the volume, defining the total aerodynamic force as

$$\vec{F}_{a,B} = \int_{Body Area} d\vec{F}_{ext} \quad (9.86)$$

and defining the propulsive thrust force to be acting forward on the vehicle as

$$\vec{F}_{p,B} = \dot{m} \vec{v}_{Ex} + \int_{Exit Area} d\vec{F}_{ext} \quad (9.87)$$

where $\int_{Exit Area} d\vec{F}_{ext}$ is known as the **pressure thrust**, one has the translational equation of motion for a variable mass vehicle as

$$m \vec{\dot{v}}_{B/N,B} = \vec{F}_{g,B} + \vec{F}_{p,B} + \vec{F}_{a,B} + 2\dot{m} \vec{x}_{Ex/B,B} + \ddot{m} \vec{x}_{Ex/B,B} \quad (9.88)$$

or in terms of body-fixed frame coordinates

$$m \begin{bmatrix} u - qw + rv \\ v - ru + pw \\ w - pv + qu \end{bmatrix} = \vec{F}_{g,B} + \vec{F}_{p,B} + \vec{F}_{a,B} + 2\dot{m} (\vec{\dot{x}}_{Ex/B,B} + \vec{\omega}_{B/N,B} \times \vec{x}_{Ex/B,B}) + \ddot{m} \vec{x}_{Ex/B,B} \quad (9.89)$$

Thus, accounting for the variable mass results in two additive terms as apparent forces in the translational equation of motion. Note that $\vec{x}_{Ex/B,B}$ changes with time as the expelled mass will alter the location of the center of mass, i.e. the origin of the body frame. These terms are often neglected due to $\vec{x}_{Ex/B}$ being small magnitude relative to the propellant exit velocity and the mass rate is roughly constant.

Continuing the rigid-body EOM analysis, at time t , one has for the inertial rotational momentum of the variable mass vehicle

$$\vec{H}_N(t) = \int_{Vol} \vec{x}_{V/N} \times \rho_V \dot{\vec{x}}_{V/N} dV \quad (9.90)$$

while at time $t + \Delta t$, the rotational momentum is

$$\begin{aligned} \vec{H}_N(t + \Delta t) &= \int_{Vol} (\vec{x}_{V/N} + \Delta \vec{x}_{V/N}) \times \rho_V (\dot{\vec{x}}_{V/N} + \Delta \dot{\vec{x}}_{V/N}) dV \\ &\quad + (\vec{x}_{Ex/N} + \Delta \vec{x}_{Ex/N}) \times \Delta m (\dot{\vec{x}}_{Ex/N} + \Delta \dot{\vec{x}}_{Ex/N}) \end{aligned} \quad (9.91)$$

represents the change in the vehicle's rotational momentum due to the change in mass, i.e. $\Delta m < 0$ for expelled mass. Then taking the limit as $\Delta t \rightarrow 0$ (and neglecting higher order Δ terms), one has

$$\dot{\vec{H}}_N = \int_{Vol} \frac{d}{dt} (\vec{x}_{V/N} \times \rho_V \dot{\vec{x}}_{V/N}) dV + \vec{x}_{Ex/N} \times \dot{m} \dot{\vec{x}}_{Ex/N} \quad (9.92)$$

Comparing this expression with Euler's rotational equation of motion, one can see that the total rate of change in the rotational momentum may be rewritten as

$$\dot{\vec{H}}_N = \int_{Vol} \vec{x}_{V/N} \times \rho_V \vec{g} dV + \int_{Area} \vec{x}_{V/N} \times d\vec{F}_{ext} + \vec{x}_{Ex/N} \dot{m} \dot{\vec{x}}_{Ex/N} \quad (9.93)$$

Next, noting $\vec{x}_{V/N} = \vec{x}_{B/N} + \vec{x}_{V/B}$, one has

$$\begin{aligned} \dot{\vec{H}}_N &= \vec{x}_{B/N} \times \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} dV + \int_{Vol} \vec{x}_{V/B} \times \rho_V \dot{\vec{x}}_{V/B} \\ &= \vec{x}_{B/N} \times \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} dV + \vec{H}_{B,N} \end{aligned} \quad (9.94)$$

where the first term is the angular momentum of the body-fixed frame and $\vec{H}_{B,N}$ is the angular momentum of the vehicle in navigation frame coordinates. Recalling

$$\vec{P}_N(t) = \int_{Vol} \rho_V \dot{\vec{x}}_{V/N} dV \quad (9.95)$$

and differentiating with respect to the navigation frame, one has

$$\dot{\vec{H}}_N = \vec{x}_{B/N} \times \vec{P}_N + \vec{x}_{B/N} \times \dot{\vec{P}}_N + \dot{\vec{H}}_{B,N} \quad (9.96)$$

Then, substituting for \vec{P}_N and $\dot{\vec{P}}_N$ from before, one has

$$\dot{\vec{H}}_N = -\vec{x}_{B/N} \times \dot{m} \vec{x}_{Ex/B} + \vec{x}_{B/N} \times \left(\int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} \dot{\vec{x}}_{Ex/N} \right) + \dot{\vec{H}}_{B,N} \quad (9.97)$$

Finally, equating Equations 9.93 and 9.97, one has

$$\begin{aligned} \dot{\vec{H}}_{B,N} - \vec{x}_{B/N} \times \dot{m} \vec{x}_{Ex/B} + \vec{x}_{B/N} \times \left(\int_{Vol} \rho_V \vec{g} dV + \int_{Area} d\vec{F}_{ext} + \dot{m} \dot{\vec{x}}_{Ex/N} \right) \\ = \int_{Vol} \vec{x}_{V/N} \times \rho_V \vec{g} dV + \int_{Area} \vec{x}_{V/N} \times d\vec{F}_{ext} + \vec{x}_{Ex/N} \dot{m} \dot{\vec{x}}_{Ex/N} \end{aligned} \quad (9.98)$$

and noting that the first mass moment about the center of mass is zero for no gravity gradient modeling and

$$\begin{aligned}\vec{x}_{V/N} &= \vec{x}_{B/N} + \vec{x}_{V/B} \\ \vec{x}_{Ex/N} &= \vec{x}_{B/N} + \vec{x}_{Ex/B} \\ \dot{\vec{x}}_{Ex/N} &= \vec{v}_{Ex} + \vec{x}_{E/B} + \dot{\vec{x}}_{B/N}\end{aligned}\quad (9.99)$$

one can rewrite the angular momentum of the variable mass vehicle about its center of mass as

$$\dot{\vec{H}}_{B,N} = \int_{Area} \vec{x}_{V/B} \times d\vec{F}_{ext} + \vec{x}_{Ex/B} \times \dot{m} (\dot{\vec{x}}_{E/B} + \vec{v}_{Ex}) \quad (9.100)$$

which after separating the external moments into aerodynamic and propulsive contributions, one has

$$\dot{\vec{H}}_{B,N} = \vec{M}_{a,N} + \vec{M}_{p,N} + \vec{x}_{Ex/B} \times \dot{m} (\dot{\vec{x}}_{E/B,B} + \vec{\omega}_{B/N} \times \vec{x}_{Ex/B}) \quad (9.101)$$

$$\dot{\vec{H}}_{B,N} = C_{N \leftarrow B} \begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} + \vec{x}_{Ex/B} \times \dot{m} (\dot{\vec{x}}_{E/B,B} + \vec{\omega}_{B/N} \times \vec{x}_{Ex/B}) \quad (9.102)$$

where $\vec{M}_p = \vec{x}_{Ex/B} \times \vec{F}_p$ and the additive triple product is known as the **jet-damping effect** because it tends to act against the vehicle's angular motion. Note that this is proportional to the vehicle's angular velocity, the square of the distance from the nozzle exit to the center of mass, and the mass flow rate. Thus, this damping is more significant for maneuvering, long vehicles with large mass flow rates, e.g. missiles.

9.3 Atmospheric Effects on Aerospace Vehicle Dynamics

For introductory FDC, one typically assumes simple atmospheric conditions, namely constant density and no wind. Furthermore, one typically simulates the flight motion of an aircraft at a particular reference altitude and assumes that the acceleration due to gravity is constant. These assumptions simplify the equations of motion as altitude does not need to be considered as state (affects density and gravity), the stability and control derivatives are constant (changes with density), the body-fixed frame velocity is equal to the airspeed velocity. This lecture will discuss how to alter the equations of motion to account for these atmospheric and variable gravity effects.

Air Density Effect and Models

In FDC, one typically includes the assumption that the aerodynamic forces and moments can be developed with a constant atmospheric density. However, if atmospheric density change is important (e.g. hypersonic flight vehicles), then one may include aerodynamic forces due to density variations as

$$\begin{aligned}X_h &= \frac{Q_\infty S_w}{\rho_\infty m} \bar{C}_X \left(\frac{\partial \rho_\infty}{\partial h} \right) \\ Y_h &= \frac{Q_\infty S_w}{\rho_\infty m} \bar{C}_Y \left(\frac{\partial \rho_\infty}{\partial h} \right) \\ Z_h &= \frac{Q_\infty S_w}{\rho_\infty m} \bar{C}_Z \left(\frac{\partial \rho_\infty}{\partial h} \right)\end{aligned}\quad (9.103)$$

while typically disregarding the effect on the moments as they are typically quite small. Here the density gradient $\partial \rho_\infty / \partial h$ is obtained from an atmospheric model. Aviation standards use the International Standard Atmosphere (ISA) uses a linear temperature vs. altitude model for different layers of the atmosphere, i.e.

$$T = T_0 + \ell(h - h_0) \quad (9.104)$$

where the linear variation, ℓ , is called **temperature lapse rate**. For the Standard Atmosphere 1962, one has

| Atmospheric Layer | Lapse Rate ℓ ($^{\circ}\text{R}/\text{ft}$) | Lower Altitude h_0 (ft) | Temperature T_0 ($^{\circ}\text{R}$) | Pressure p_0 (psf) | Density ρ_0 (sl/ft 3) |
|-------------------|--|---------------------------|--|----------------------|--------------------------------|
| Troposphere | -3.5662×10^{-3} | 0 | 518.69 | 2,116.2 | 2.3769×10^{-3} |
| Stratosphere I | 0 | 36,089 | 389.99 | 472.68 | 7.0613×10^{-4} |
| Stratosphere II | 5.4864×10^{-4} | 65,617 | 389.99 | 114.35 | 1.7083×10^{-4} |

Using this information, as well as the **aerostatic equation**

$$\frac{dp}{dh} = -\rho g_0 \quad (9.105)$$

where g_0 is the acceleration due to gravity at mean sea level (MSL), and the **perfect-gas equation**

$$p = \rho RT \quad (9.106)$$

where R is the universal gas constant for air, $1716.5 \text{ ft}^2/(\text{s}^2 - ^{\circ}\text{R})$, one can form the following models for the different strata of the atmosphere. The troposphere mathematical model is

$$\begin{aligned} T &= 518.69 - (3.5662 \times 10^{-3})h \quad ^{\circ}\text{R} \\ p &= (1.1376 \times 10^{-11})T^{5.256} \quad \text{psf} \\ \rho &= (6.6277 \times 10^{-15})T^{4.256} \quad \text{sl/ft}^3 \end{aligned} \quad (9.107)$$

The stratosphere I mathematical model is

$$\begin{aligned} T &= 389.99 \quad ^{\circ}\text{R} \\ p &= (2678.4) \exp\left((-4.8063 \times 10^{-5})h\right) \quad \text{psf} \\ \rho &= (1.4939 \times 10^{-6})p \quad \text{sl/ft}^3 \end{aligned} \quad (9.108)$$

The stratosphere II mathematical model is

$$\begin{aligned} T &= 389.99 + (5.4864 \times 10^{-4})(h - 65617) \quad ^{\circ}\text{R} \\ p &= (3.7930 \times 10^{90})T^{-34.164} \quad \text{psf} \\ \rho &= (2.2099 \times 10^{87})T^{-35.164} \quad \text{sl/ft}^3 \end{aligned} \quad (9.109)$$

An approximation for the density that is often used specifically for these models is

$$\begin{aligned} \text{Troposphere: } \rho &= (2.3769 \times 10^{-3}) \exp\left(\frac{-h}{29,730}\right) \quad \text{sl/ft}^3 \\ \text{Stratosphere I: } \rho &= (7.0613 \times 10^{-4}) \exp\left(\frac{h - 36,089}{20,806}\right) \quad \text{sl/ft}^3 \\ \text{Stratosphere II: } \rho &= (1.7083 \times 10^{-4}) \exp\left(\frac{-(h - 65,617)}{29,730}\right) \quad \text{sl/ft}^3 \end{aligned} \quad (9.110)$$

for which the density-altitude gradient can be easily calculated as a function of h .

Wind Effects and Models

In deriving the equations of motion in introductory FDC, one typically uses the stability frame for coordinated flight where the velocity vector is colinear with the body-fixed frame x_B -axis, i.e. $\bar{u} = v_a$, and one can use approximations for Δv and Δw by β and α . However, in the presence of wind, this relationship is more complex due to the wind triangle which can be expressed as

$$\vec{v}_{B/N} = \vec{v}_\infty + \vec{v}_{W/N} \quad (9.111)$$

where $\vec{v}_{B/N}$ is the groundspeed vector, i.e. the velocity of the body-fixed frame relative to the “inertial” navigation frame, $\vec{v}_\infty = \vec{v}_{B/W}$ is the airspeed vector, i.e. the velocity of the body-fixed frame relative to the air mass, and $\vec{v}_{W/N}$ is the wind speed vector, i.e. the velocity of the air mass relative to the “inertial” navigation frame. In FDC, as the aerodynamic forces and moments are typically a function of the airspeed velocity instead of the groundspeed vector, it is often more useful to use the airspeed velocity vector as part of the state vector in the equations of motion.

First, utilizing navigation frame and body-fixed frame coordinates, one can write

$$\vec{v}_{B/N,N} = C_{N \leftarrow B} \vec{v}_{\infty,B} + \vec{v}_{W/N,N} \quad (9.112)$$

and differentiating, one has

$$\dot{\vec{v}}_{B/N,N} = C_{N \leftarrow B} \dot{\vec{v}}_{\infty,B} + C_{N \leftarrow B} [\vec{\omega}_{B/N,B}]_\times \vec{v}_{\infty,B} + \dot{\vec{v}}_{W/N,N} \quad (9.113)$$

Next, recall the sum of forces in “inertial” navigation frame coordinates can be written as

$$\vec{F}_{a,N} + \vec{F}_{p,N} + \vec{F}_{g,N} = m \dot{\vec{v}}_{B/N,N} \quad (9.114)$$

Then, substituting and converting to body-fixed frame coordinates, one has

$$\vec{F}_{a,B} + \vec{F}_{p,B} + m C_{B \leftarrow N} \vec{g}_N = m \left(\dot{\vec{v}}_{\infty,B} + [\vec{\omega}_{B/N,B}]_\times \vec{v}_{\infty,B} + C_{B \leftarrow N} \dot{\vec{v}}_{W/N,N} \right) \quad (9.115)$$

or via mass normalization and using $\vec{v}_{\infty,B} = [u \ v \ w]^T$

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} + \begin{bmatrix} \dot{u}_{W/N,N} \\ \dot{v}_{W/N,N} \\ \dot{w}_{W/N,N} \end{bmatrix} \quad (9.116)$$

Note that if one has **steady wind**, i.e. $\dot{\vec{v}}_{W/N,N} = 0$, then these equations have the same mathematical form as the no-wind translation EOMs. Furthermore, one can integrate these velocity equations of motion to find the vehicle’s inertial position as

$$\vec{x}_{B/N,N} = \int \vec{v}_{B/N,N} dt = \int C_{N \leftarrow B}(t) \vec{v}_{\infty,B}(t) + \vec{v}_{W/N,N} dt \quad (9.117)$$

Thus, one can simply use the airspeed vector components in the EOMs which would add a positional offset that grows linearly with time due to the steady wind.

For assessing the effects of **wind shear** for $3 \text{ ft} < h < 1000 \text{ ft}$ as

$$\bar{u}_{ws} = \|\vec{v}_{w,20}\|_2 \frac{\log(hz_0^{-1})}{\log(20z_0^{-1})} \quad (9.118)$$

with \bar{u}_{ws} as the mean wind speed in the navigation frame, $\vec{v}_{w,20}$ as the measured wind speed at a height above ground level of 20 feet, $z_0 = 0.15$ feet for take-off, approach, and landing flight phases and $z_0 = 2$ feet for all other flight phases.

For assessing the effects of **gusts**, i.e., the **unsteady wind component**, on the velocity EOMs as opposed to the constant steady wind component, one typically assumes that the unsteady wind component is a zero-mean **random gust** which take on some random properties of varying magnitude denoted as \vec{v}_g . For simplicity, assume one has no steady-wind, i.e. one can redefine the body-fixed frame velocity as

$$\vec{v}_{B/N} = \vec{v}_\infty + \vec{v}_g \quad (9.119)$$

which uses both $\vec{v}_\infty = [u \ v \ w]^T$ and $\vec{v}_g = [u_g \ v_g \ w_g]^T$ in the state vector for the EOMs. Thus, by component one has

$$\begin{bmatrix} u_{tot} \\ v_{tot} \\ w_{tot} \end{bmatrix} = \begin{bmatrix} u + u_g \\ v + v_g \\ w + w_g \end{bmatrix} \quad (9.120)$$

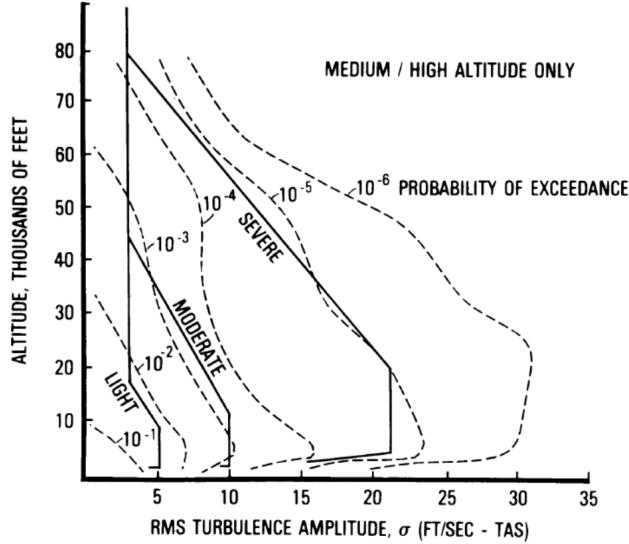
where u , v , and w implicitly model the velocity relative to a hypothetically *still* air mass. Two of the most widely used random gust models are the **Dryden gust model** and the **von Kármán gust model**. The Dryden gust state equations are provided here for addition to the stability frame coordinates as

$$\begin{bmatrix} \dot{u}_g(t) \\ \dot{v}_g(t) \\ \dot{v}_{g1}(t) \\ \dot{w}_g(t) \\ \dot{w}_{g1}(t) \end{bmatrix} = \begin{bmatrix} -\frac{\bar{u}}{L_u} & 0 & 0 & 0 & 0 \\ 0 & -\frac{\bar{u}}{L_u} & \sigma_v(1 - \sqrt{3}) \left(\frac{\bar{u}}{L_v}\right)^{3/2} & 0 & 0 \\ 0 & 0 & -\frac{\bar{u}}{L_v} & 0 & 0 \\ 0 & 0 & 0 & -\frac{\bar{u}}{L_w} & \sigma_w(1 - \sqrt{3}) \left(\frac{\bar{u}}{L_w}\right)^{3/2} \\ 0 & 0 & 0 & 0 & -\frac{\bar{u}_a}{L_w} \end{bmatrix} \begin{bmatrix} u_g(t) \\ v_g(t) \\ v_{g1}(t) \\ w_g(t) \\ w_{g1}(t) \end{bmatrix} + \begin{bmatrix} \sigma_u \left(\frac{2\bar{u}}{\pi L_u}\right)^{1/2} \\ \sigma_v \left(\frac{3\bar{u}}{L_v}\right)^{1/2} \\ 1 \\ \sigma_w \left(\frac{3\bar{u}}{L_w}\right)^{1/2} \\ 1 \end{bmatrix} \vec{n}(t)$$

$$\dot{\vec{x}}_g = A_g \vec{x}_g + B_g n \quad (9.121)$$

where the driving function, $n(t)$, is a zero-mean, additive white Gaussian noise (AWGN) of unit intensity across all five channels, $[L_u \ L_v \ L_w]$ are $[h \ 145h^{1/3} \ 145h^{1/3}]$ for $h < 1750 \text{ ft}$ and $[1750 \ 1750 \ 1750]$ for $h \geq 1750 \text{ ft}$, and σ_u , σ_v , and σ_w are the standard deviations of the gusts, or **RMS gust intensities** which can be obtained from data such as “MIL-F-8785C Military Specification: Flying Qualities of Piloted Airplanes”

which provides the plot of three levels of RMS gust intensities for different altitudes: *light*, *moderate*, and *severe*.



Note that here the **probability of exceedance** is the probability that the RMS gust intensity would exceed the value shown on the curves at that altitude. Also note that for numerical simulations, $n(t)$ can be approximated as continuous over a short time step of size Δt , thus approximating $n(t)$ by a random sequence n_i which are zero-mean Gaussian with variance $1/\Delta t$.

Lastly, using the wind frame Euler angles, i.e. the angle of attack and sideslip angles, to transform the instantaneous velocity magnitude, v_∞ , to body-fixed frame coordinates, one has

$$\begin{bmatrix} v_\infty \cos \alpha_{tot} \cos \beta_{tot} \\ v_\infty \sin \beta_{tot} \\ v_\infty \sin \alpha_{tot} \cos \beta_{tot} \end{bmatrix} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} + \begin{bmatrix} u_g \\ v_g \\ w_g \end{bmatrix} \quad (9.122)$$

From this equation, one can form the following relationships. First, the magnitude equation can be written as

$$v_\infty^2 = (u + u_g)^2 + (v + v_g)^2 + (w + w_g)^2 \quad (9.123)$$

Second, taking the third row divided by the first row, one has

$$\tan \alpha_{tot} = \frac{w + w_g}{u + u_g} \quad (9.124)$$

Third, the second row can be rewritten as

$$\sin \beta_{tot} = \frac{v + v_g}{v_\infty} \quad (9.125)$$

For small angles and $u_w \ll u$, one can write

$$\alpha_{tot} \approx \frac{w}{u} + \frac{w_g}{u} = \alpha + \alpha_g \quad (9.126)$$

$$\beta_{tot} \approx \frac{v}{v_\infty} + \frac{v_g}{v_a} = \beta + \beta_g \quad (9.127)$$

and similarly

$$\dot{\alpha}_{tot} \approx \dot{\alpha} + \dot{\alpha}_g \quad (9.128)$$

which are often used in linear flight dynamics instead of v_{tot} and w_{tot} .

With these relationships and a simplifying assumption, it is easy to model the aerodynamic forces and moments using the total velocity components. The assumption is that the axial and lateral gusts velocities have a negligible variation across the flight vehicle compared to the global variations of the gusts relative to the surface of the Earth. Note that this does not assume anything about the vertical gusts due to the presence of $\dot{\alpha}$. Then, one may simply use the addition formulas for $u + u_g$, $v + v_g$ (or $\approx \beta + \beta_g$), $w + w_g$ (or $\approx \alpha + \alpha_g$), and $\dot{\alpha} + \dot{\alpha}_g$ for computing the stability derivative contributions with respect to u , v (or β), w (or α), and \dot{w} (or $\dot{\alpha}$) for X , Y , Z , L , M , and N .

Furthermore, for the linearized flight dynamics, if one can uses the linear Dryden gust model above combined with the nominal LTI state-space state equation, i.e.

$$\Delta \dot{\vec{x}} = A \Delta \vec{x} + B \Delta \vec{u} \quad (9.129)$$

where $\Delta \vec{x} = [\Delta u \ \Delta \beta \ \Delta \alpha \ \Delta p \ \Delta q \ \Delta r \ \Delta \phi \ \Delta \theta \ \Delta \psi]^T$ and $\Delta \vec{u} = [\Delta \delta_a \ \Delta \delta_e \ \Delta \delta_r \ \Delta \delta_t]^T$, then one can form the following augmented LTI state equation

$$\begin{bmatrix} \Delta \dot{\vec{x}} \\ \dot{\vec{x}}_g \end{bmatrix} = \begin{bmatrix} A & G_g C_g \\ 0 & A_g \end{bmatrix} \begin{bmatrix} \Delta \vec{x} \\ \vec{x}_g \end{bmatrix} + \begin{bmatrix} B & G_g D_g \\ 0 & B_g \end{bmatrix} \begin{bmatrix} \Delta \vec{u} \\ n \end{bmatrix} \quad (9.130)$$

where the gust output matrix and feedthrough matrix can be constructed as

$$C_g = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{\bar{u}} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{\bar{u}} & 0 \\ 0 & 0 & 0 & -\frac{1}{L_w} & \sigma_w (1 - \sqrt{3}) \left(\frac{\bar{u}}{L_w^3} \right)^{1/2} \end{bmatrix} \quad (9.131)$$

and

$$D_g = \begin{bmatrix} 0 \\ 0 \\ 0 \\ \sigma_w \left(\frac{3}{L_w \bar{u}} \right)^{1/2} \end{bmatrix} \quad (9.132)$$

to output u_g , β_g , α_g , and $\dot{\alpha}_g$ from the gust state vector $\vec{x}_g = [u_g \ v_g \ v_{g1} \ w_g \ w_{g1}]^T$, and the mapping matrix

from these gust outputs to the stability derivatives

$$G_g = \begin{bmatrix} X_u + \frac{X_{\dot{\alpha}}Z_u}{\bar{u}-Z_{\dot{\alpha}}} & 0 & X_{\alpha} + \frac{X_{\dot{\alpha}}Z_{\alpha}}{\bar{u}-Z_{\dot{\alpha}}} & X_{\dot{\alpha}} + \frac{X_{\dot{\alpha}}Z_{\dot{\alpha}}}{\bar{u}-Z_{\dot{\alpha}}} \\ 0 & \frac{Y_{\beta}}{\bar{u}} & 0 & 0 \\ \frac{Z_u}{\bar{u}-Z_{\dot{\alpha}}} & 0 & \frac{Z_{\alpha}}{\bar{u}-Z_{\dot{\alpha}}} & \frac{Z_{\dot{\alpha}}}{\bar{u}-Z_{\dot{\alpha}}} \\ 0 & L_{\beta}^* & 0 & 0 \\ M_u + \frac{M_{\dot{\alpha}}Z_u}{\bar{u}-Z_{\dot{\alpha}}} & 0 & M_{\alpha} + \frac{M_{\dot{\alpha}}Z_{\alpha}}{\bar{u}-Z_{\dot{\alpha}}} & M_{\dot{\alpha}} + \frac{M_{\dot{\alpha}}Z_{\dot{\alpha}}}{\bar{u}-Z_{\dot{\alpha}}} \\ 0 & N_{\beta}^* & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (9.133)$$

Alternatively, this can also be used with the nonlinear state equation as an additive linear portion.

References

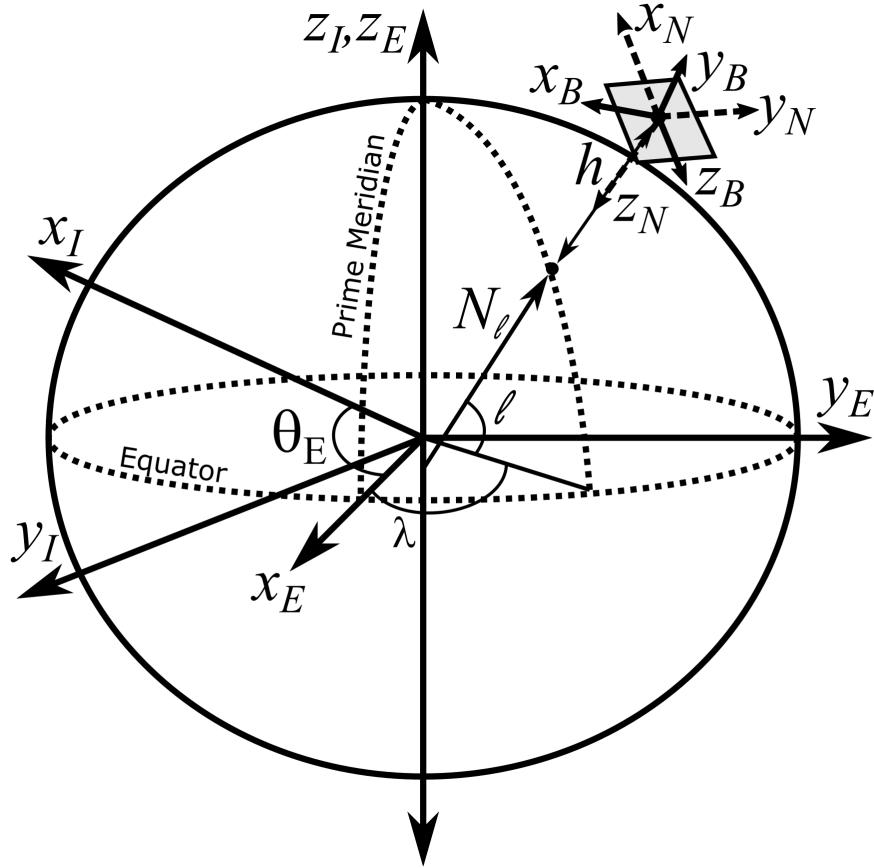
For more information, please refer to the following

- Schmidt, D. K., “Appendix C Models of Atmospheric Turbulence,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 839-851

9.4 Earth Frame Effects on Aerospace Vehicle Dynamics

In introductory FDC, one typically assumes that the navigation frame axes (e.g. North, East, and Down) can be treated as inertial, i.e. a “flat-Earth” approximation model. However, for vehicles that operate at high supersonic velocities and/or fly long distances, this assumption may not have a negligible effect. Thus, this section will discuss the effects of modeling the Earth frame as rotating about a constant axis for an ellipsoidal shape, i.e. the **reference ellipsoid** which adds to the apparent forces and moments of the vehicle. This notably does not discuss the centripetal acceleration of the Earth about the Sun nor the Sun about the Milky Way. As discussed previously, the Earth is truly a non-analytical geoid and has a very slight wobble in its rotation axis. Note that this section will look at the changes to the EOMs for a rigid flight vehicle with constant mass, no rotating mass, and no wind. However, these effects could also be added to the equation.

Assuming this model of the Earth, consider the relationship between the ECI frame (subscript *I*), the ECEF frame (subscript *E*), geodetic coordinates (ℓ, λ, h), the navigation frame (subscript *N*), and the body-fixed frame (subscript *B*) as shown in the following diagram.



where $\theta_E = \omega_E t$ is the rotation angle between the ECI and ECEF frames, ω_E is defined as 7.2921150×10^{-5} rad/s by the WGS 84, t is referenced from some reference point, e.g. the Geocentric Celestial Reference Frame (GCRF) ECI frame is referenced at 12:00 Terrestrial Time on January 1, 2000 with the ITRF ECEF frame, and N_ℓ is the **Earth's prime-vertical radius of curvature** of the reference ellipsoid which depends on the latitude as

$$N_\ell = \frac{R_e}{\sqrt{1 - e_E^2 \sin^2 \ell}} \quad (9.134)$$

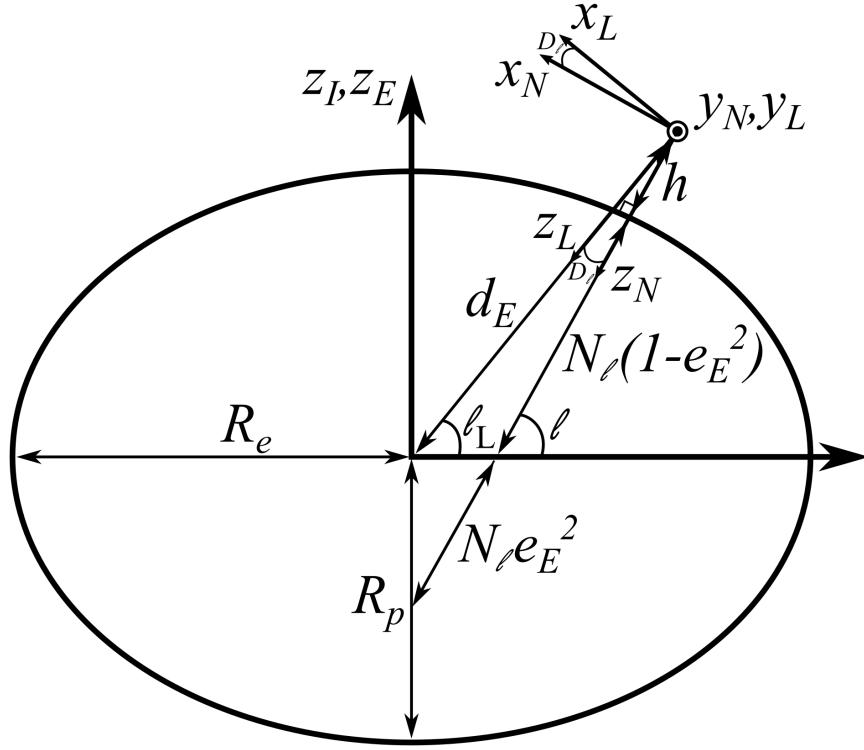
where R_e is the equatorial radius of the reference ellipsoid defined as 6,378,137.0 m by the WGS-84 and e_E is the eccentricity of the reference ellipsoid defined as

$$e_E = \sqrt{f_E(2 - f_E)} = 0.081819190842622 \quad (9.135)$$

where f_E is the **flattening** of the reference ellipsoid defined as 1/298.257223563 by the WGS-84. Furthermore, R_p is the reference ellipsoid's polar radius defined as

$$R_p = R_e(1 - f_E) = 6,356,752.3 \text{ m} \quad (9.136)$$

Notably, as shown in the following diagram.



where d_E is the geocentric distance from Earth's center, N_ℓ has notably been split into two convenient sections, and the **deviation of the normal**, D_ℓ , is related to the geodetic latitude, ℓ , and the geocentric latitude, ℓ_L , by

$$\ell = \ell_L + D_\ell \quad (9.137)$$

This angle also defines the rotation between the local-vertical, local-horizontal (LVLH) frame (subscript L) and the navigation frame. Thus, for a **spherical-Earth model**, i.e. $D_\ell = 0$, and $\ell = \ell_L$, the LVLH and navigation frames are the same.

Furthermore, using

$$n_\ell = \frac{e_E^2 N_\ell}{N_\ell + h} \quad (9.138)$$

one can show

$$\tan D_\ell = \frac{n_\ell \sin \ell \cos \ell}{1 - n_\ell \sin^2 \ell} = \frac{n_\ell \sin \ell_e \cos \ell_L}{1 - n_\ell \cos^2 \ell_L} \quad (9.139)$$

$$\sin \ell_L = \frac{(1 - n_\ell) \sin \ell}{\sqrt{1 - n_\ell(2 - n_\ell) \sin^2 \ell}} \quad (9.140)$$

$$\cos \ell_L = \frac{\cos \ell}{\sqrt{1 - n_\ell(2 - n_\ell) \sin^2 \ell}} \quad (9.141)$$

$$\tan \ell_L = (1 - n_\ell) \tan \ell \quad (9.142)$$

Lastly, the geocentric distance is related to the geodetic height via

$$d_E = (N_\ell + h) \sqrt{1 - n(2 - n) \sin^2 \ell} \quad (9.143)$$

Geodetic and Geocentric Rates of Change

Through this diagram, the position of the vehicle's center of mass, written in ECEF coordinates can be related to the geocentric or geodetic coordinates as

$$\vec{x}_{B/E,E} = \begin{bmatrix} d_E \cos \ell_L \cos \lambda \\ d_E \cos \ell_L \sin \lambda \\ d_E \sin \ell_L \end{bmatrix} = \begin{bmatrix} (N_\ell + h) \cos \lambda \cos \ell \\ (N_\ell + h) \sin \lambda \cos \ell \\ (N_\ell(1 - e_E^2) + h) \sin \ell \end{bmatrix} \quad (9.144)$$

Furthermore, the change in the coordinates as related to geocentric coordinates can be written as

$$\dot{\vec{x}}_{B/E,E} = \begin{bmatrix} -\dot{\ell}_L d_E \sin \ell_L \cos \lambda - \dot{\lambda} d_E \cos \ell_L \sin \lambda + \dot{d}_E \cos \ell_L \cos \lambda \\ -\dot{\ell}_L d_E \sin \ell_L \sin \lambda + \dot{\lambda} d_E \cos \ell_L \cos \lambda + \dot{d}_E \cos \ell_L \sin \lambda \\ \dot{\ell}_L d_E \cos \ell_L + \dot{d}_E \sin \ell_L \end{bmatrix} \quad (9.145)$$

or written in LVLH frame coordinates, one has

$$\dot{\vec{x}}_{B/E,L} = C_{L \leftarrow E} \begin{bmatrix} -\dot{\ell}_L d_E \sin \ell_L \cos \lambda - \dot{\lambda} d_E \cos \ell_L \sin \lambda + \dot{d}_E \cos \ell_L \cos \lambda \\ -\dot{\ell}_L d_E \sin \ell_L \sin \lambda + \dot{\lambda} d_E \cos \ell_L \cos \lambda + \dot{d}_E \cos \ell_L \sin \lambda \\ \dot{\ell}_L d_E \cos \ell_L + \dot{d}_E \sin \ell_L \end{bmatrix} \quad (9.146)$$

where the rotation matrix is obtained via

$$C_{L \leftarrow E} = C_2(-\ell_L - \pi/2) C_3(\lambda) = \begin{bmatrix} -\sin \ell_L \cos \lambda & -\sin \ell_L \sin \lambda & \cos \ell_L \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \ell_L \cos \lambda & -\cos \ell_L \sin \lambda & -\sin \ell_L \end{bmatrix} \quad (9.147)$$

or

$$\dot{\vec{x}}_{B/E,L} = \begin{bmatrix} \dot{\ell}_L d_E \\ \dot{\lambda} d_E \cos \ell_L \\ -\dot{d}_E \end{bmatrix} \quad (9.148)$$

However, defining the velocity of the vehicle's center of mass in body-fixed frame coordinates, one has

$$\dot{\vec{x}}_{B/E,L} = C_{L \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (9.149)$$

which provides the **geocentric rates supplementary equation**

$$\begin{bmatrix} \dot{\ell}_L d_E \\ \dot{\lambda} d_E \cos \ell_L \\ -\dot{d}_E \end{bmatrix} = C_{L \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (9.150)$$

or

$$\begin{bmatrix} \dot{\ell}_L \\ \dot{\lambda} \\ \dot{d}_E \end{bmatrix} = \begin{bmatrix} \frac{1}{d_E} & 0 & 0 \\ 0 & \frac{1}{d_E \cos \ell_L} & 0 \\ 0 & 0 & -1 \end{bmatrix} C_{L \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (9.151)$$

Alternatively, for geodetic coordinates, one can show

$$\frac{d}{dt} ((N_\ell + h) \cos \ell) = -(M_\ell + h) \sin \ell \quad (9.152)$$

and

$$\frac{d}{dt} \left((N_\ell (1 - e_{Earth}^2) + h) \sin \ell \right) = (M_\ell + h) \cos \ell \quad (9.153)$$

where M_ℓ is the **Earth's meridinal radius of curvature** as

$$M_\ell = \frac{R_E (1 - e_E^2)}{\left(1 - e_E^2 \sin^2 \ell\right)^{3/2}} \quad (9.154)$$

Thus, alternatively the velocity of this point can be written using geodetic coordinates as

$$\frac{d}{dt} \vec{x}_{B/E,E} = \begin{bmatrix} -\dot{\ell}(M_\ell + h) \sin \ell \cos \lambda - \dot{\lambda}(N_\ell + h) \cos \ell \sin \lambda + \dot{h} \cos \ell \cos \lambda \\ -\dot{\ell}(M_\ell + h) \sin \ell \sin \lambda + \dot{\lambda}(N_\ell + h) \cos \ell \cos \lambda + \dot{h} \cos \ell \sin \lambda \\ \dot{\ell}(M_\ell + h) \cos \ell + \dot{h} \sin \ell \end{bmatrix} \quad (9.155)$$

or written in navigation frame coordinates, one has

$$\frac{d}{dt} \vec{x}_{B/E,N} = C_{N \leftarrow E} \begin{bmatrix} -\dot{\ell}(M_\ell + h) \sin \ell \cos \lambda - \dot{\lambda}(N_\ell + h) \cos \ell \sin \lambda + \dot{h} \cos \ell \cos \lambda \\ -\dot{\ell}(M_\ell + h) \sin \ell \sin \lambda + \dot{\lambda}(N_\ell + h) \cos \ell \cos \lambda + \dot{h} \cos \ell \sin \lambda \\ \dot{\ell}(M_\ell + h) \cos \ell + \dot{h} \sin \ell \end{bmatrix} \quad (9.156)$$

where

$$C_{N \leftarrow E} = C_2(-\ell - \pi/2) C_3(\lambda) = \begin{bmatrix} -\sin \ell \cos \lambda & -\sin \ell \sin \lambda & \cos \ell \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \ell \cos \lambda & -\cos \ell \sin \lambda & -\sin \ell \end{bmatrix} \quad (9.157)$$

or

$$\frac{d}{dt} \vec{x}_{B/E,N} = \begin{bmatrix} \dot{\ell}(M_\ell + h) \\ \dot{\lambda}(N_\ell + h) \cos \ell \\ -\dot{h} \end{bmatrix} \quad (9.158)$$

which provides the **geodetic rates supplementary equation**

$$\begin{bmatrix} \dot{\ell} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{M_\ell + h} & 0 & 0 \\ 0 & \frac{1}{(N_\ell + h) \cos \ell} & 0 \\ 0 & 0 & -1 \end{bmatrix} C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (9.159)$$

Rotation Equation of Motion Effects

The angular velocity effect of the rotating Earth will appear as two additional angular velocity terms for the ECI and ECEF frames, i.e.

$$\vec{\omega}_{B/I} = \vec{\omega}_{B/L} + \vec{\omega}_{L/E} + \vec{\omega}_{E/I} = \vec{\omega}_{B/N} + \vec{\omega}_{N/E} + \vec{\omega}_{E/I} \quad (9.160)$$

Thus, for the “flat-Earth” model, one considers both the angular velocities of the ECEF frame relative to the ECI frame, $\vec{\omega}_{E/I}$, and the LVLH/navigation frames relative to the ECEF frame, $\vec{\omega}_{L/E}/\vec{\omega}_{N/E}$, to be zero.

For $\vec{\omega}_{E/I}$, by definition of the ECI frame, one has

$$\vec{\omega}_{E/I,E} = \begin{bmatrix} 0 \\ 0 \\ \omega_E \end{bmatrix} \quad (9.161)$$

and the rotation from the ECI frame to the ECEF frame is given by

$$C_{E \leftarrow I} = \begin{bmatrix} \cos \omega_E t & \sin \omega_E t & 0 \\ -\sin \omega_E t & \cos \omega_E t & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos \theta_E & \sin \theta_E & 0 \\ -\sin \theta_E & \cos \theta_E & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (9.162)$$

where ω_E is

This can be written in LVLH frame coordinates as

$$\vec{\omega}_{E/I,L} = \begin{bmatrix} \omega_E \cos \ell_L \\ 0 \\ -\omega_E \sin \ell_L \end{bmatrix} \quad (9.163)$$

or navigation frame coordinates as

$$\vec{\omega}_{E/I,N} = \begin{bmatrix} \omega_E \cos \ell \\ 0 \\ -\omega_E \sin \ell \end{bmatrix} \quad (9.164)$$

and in body-fixed frame coordinates as

$$\vec{\omega}_{E/I,B} = C_{B \leftarrow L}(\phi_L, \theta_L, \psi_L) \begin{bmatrix} \omega_E \cos \ell_L \\ 0 \\ -\omega_E \sin \ell_L \end{bmatrix} = C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} \omega_E \cos \ell \\ 0 \\ -\omega_E \sin \ell \end{bmatrix} \quad (9.165)$$

where $(\phi_L, \theta_L, \psi_L)$ are the $3 - 2 - 1$ Euler angles from the LVLH frame to the body-fixed frame and (ϕ, θ, ψ) are the $3 - 2 - 1$ Euler angles from the navigation frame to the body-fixed frame. Note that for flat-Earth and spherical-Earth models, these are the same.

For $\vec{\omega}_{L/E}$, one can use the definition of the derivative of a rotation matrix, i.e.

$$\dot{C}_{E \leftarrow L} = C_{E \leftarrow L} [\vec{\omega}_{L/E,L}]_\times \quad (9.166)$$

Thus, one has

$$\begin{aligned} & \begin{bmatrix} \dot{\lambda} \sin \ell_L \sin \lambda - \dot{\ell}_L \cos \ell_L \cos \lambda & -\dot{\lambda} \sin \ell_L \cos \lambda - \dot{\ell}_L \cos \ell_L \sin \lambda & -\dot{\ell}_L \sin \ell_L \\ -\dot{\lambda} \cos \lambda & -\dot{\lambda} \sin \lambda & 0 \\ \dot{\lambda} \cos \ell_L \sin \lambda + \dot{\ell}_L \sin \ell_L \cos \lambda & -\dot{\lambda} \cos \ell_L \cos \lambda + \dot{\ell}_L \sin \ell_L \sin \lambda & -\dot{\ell}_L \cos \ell_L \end{bmatrix} \\ & = \begin{bmatrix} -\sin \ell_L \cos \lambda & -\sin \ell_L \sin \lambda & \cos \ell_L \\ -\sin \lambda & \cos \lambda & 0 \\ -\cos \ell_L \cos \lambda & -\cos \ell_L \sin \lambda & -\sin \ell_L \end{bmatrix} [\vec{\omega}_{L/E,L}]_{\times} \end{aligned} \quad (9.167)$$

from which it can be shown that

$$\vec{\omega}_{L/E,L} = \begin{bmatrix} \dot{\lambda} \cos \ell_L \\ -\dot{\ell}_L \\ -\dot{\lambda} \sin \ell_L \end{bmatrix} \quad (9.168)$$

Thus,

$$\vec{\omega}_{L/I,L} = \begin{bmatrix} (\omega_E + \dot{\lambda}) \cos \ell_L \\ -\dot{\ell}_L \\ -(\omega_E + \dot{\lambda}) \sin \ell_L \end{bmatrix} \quad (9.169)$$

Similarly, for $\vec{\omega}_{N/E}$, one has

$$\vec{\omega}_{N/E,N} = \begin{bmatrix} \dot{\lambda} \cos \ell \\ -\dot{\ell} \\ -\dot{\lambda} \sin \ell \end{bmatrix} \quad (9.170)$$

and

$$\vec{\omega}_{N/I,N} = \begin{bmatrix} (\omega_E + \dot{\lambda}) \cos \ell \\ -\dot{\ell} \\ -(\omega_E + \dot{\lambda}) \sin \ell \end{bmatrix} \quad (9.171)$$

Finally, recall that

$$\vec{\omega}_{B/L,B} = \begin{bmatrix} p_{B/L,B} \\ q_{B/L,B} \\ r_{B/L,B} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta_L \\ 0 & \cos \phi_L & -\sin \phi_L \cos \theta_L \\ 0 & -\sin \phi_L & \cos \phi_L \cos \theta_L \end{bmatrix} \begin{bmatrix} \dot{\phi}_L \\ \dot{\theta}_L \\ \dot{\psi}_L \end{bmatrix} \quad (9.172)$$

and

$$\vec{\omega}_{B/N,B} = \begin{bmatrix} p_{B/N,B} \\ q_{B/N,B} \\ r_{B/N,B} \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (9.173)$$

which are equivalent for a spherical-Earth model and is the “inertial” angular velocity for a flat-Earth model.

Thus, to account for the rotation of the Earth, one must redefine the new inertial angular velocity coordinates as

$$\begin{aligned} \vec{\omega}_{B/I,B} &= \begin{bmatrix} 1 & 0 & -\sin \theta_L \\ 0 & \cos \phi_L & -\sin \phi_L \cos \theta_L \\ 0 & -\sin \phi_L & \cos \phi_L \cos \theta_L \end{bmatrix} \begin{bmatrix} \dot{\phi}_L \\ \dot{\theta}_L \\ \dot{\psi}_L \end{bmatrix} + C_{B \leftarrow L}(\phi_L, \theta_L, \psi_L) \begin{bmatrix} (\omega_E + \dot{\lambda}) \cos \ell_L \\ -\dot{\ell}_L \\ -(\omega_E + \dot{\lambda}) \sin \ell_L \end{bmatrix} \\ \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} p_{B/L,B} \\ q_{B/L,B} \\ r_{B/L,B} \end{bmatrix} + \begin{bmatrix} p_{L/I,B} \\ q_{L/I,B} \\ r_{L/I,B} \end{bmatrix} \end{aligned} \quad (9.174)$$

or

$$\begin{aligned}\vec{\omega}_{B/I,B} &= \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & -\sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} (\omega_E + \dot{\lambda}) \cos\ell \\ -\dot{\ell} \\ -(\omega_E + \dot{\lambda}) \sin\ell \end{bmatrix} \\ \begin{bmatrix} p \\ q \\ r \end{bmatrix} &= \begin{bmatrix} p_{B/N,B} \\ q_{B/N,B} \\ r_{B/N,B} \end{bmatrix} + \begin{bmatrix} p_{N/I,B} \\ q_{N/I,B} \\ r_{N/I,B} \end{bmatrix}\end{aligned}\quad (9.175)$$

Furthermore, with this new inertial angular acceleration, the *rotation equation of motion* remains the same as before, i.e.

$$\begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}}(\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}}(r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}}(\dot{p} - qr) \end{bmatrix} \quad (9.176)$$

However, the relationship between the different rotation matrices must be used to find the LVLH-to-body-fixed frame Euler angle rates as the *supplemental equation*,

$$\begin{bmatrix} \dot{\phi}_L \\ \dot{\theta}_L \\ \dot{\psi}_L \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi_L \tan\theta_L & \cos\phi_L \tan\theta_L \\ 0 & \cos\phi_L & -\sin\phi_L \\ 0 & \sin\phi_L \sec\theta_L & \cos\phi_L \sec\theta_L \end{bmatrix} \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} p_{L/I,B} \\ q_{L/I,B} \\ r_{L/I,B} \end{bmatrix} \right) \quad (9.177)$$

or the navigation-to-body-fixed frame Euler angle rates as

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin\phi \tan\theta & \cos\phi \tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi \sec\theta & \cos\phi \sec\theta \end{bmatrix} \left(\begin{bmatrix} p \\ q \\ r \end{bmatrix} - \begin{bmatrix} p_{N/I,B} \\ q_{N/I,B} \\ r_{N/I,B} \end{bmatrix} \right) \quad (9.178)$$

Translation Equation of Motion Effects

Recall the position of the vehicle's center of mass with respect to the center of Earth, i.e. the ECEF/ECI frame origin, can be written as

$$\vec{x}_{B/E,E} = \begin{bmatrix} d_E \cos\ell_L \cos\lambda \\ d_E \cos\ell_L \sin\lambda \\ d_E \sin\ell_L \end{bmatrix} = \begin{bmatrix} (N_\ell + h) \cos\lambda \cos\ell \\ (N_\ell + h) \sin\lambda \cos\ell \\ (N_\ell(1 - e_E^2) + h) \sin\ell \end{bmatrix} \quad (9.179)$$

and in ECI frame coordinates, one has

$$\vec{x}_{B/I,I} = C_{I \leftarrow E} \vec{x}_{B/E,E} \quad (9.180)$$

Differentiating this and relating this to body-fixed frame coordinates, one has

$$\dot{\vec{x}}_{B/I,I} = C_{I \leftarrow E} \dot{\vec{x}}_{B/E,E} + \dot{C}_{I \leftarrow E} \vec{x}_{B/E,E} \quad (9.181)$$

However, by definition

$$\dot{\vec{x}}_{B/E,E} = C_{E \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (9.182)$$

Thus, one has

$$\dot{\vec{x}}_{B/I,I} = C_{I \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{I \leftarrow E} [\vec{\omega}_{E/I,E}]_x \dot{\vec{x}}_{B/E,E} \quad (9.183)$$

and taking the derivative for the inertial acceleration and assuming $\dot{\omega}_{E/I,E} = 0$, one has

$$\begin{aligned} \ddot{\vec{x}}_{B/I,I} &= C_{I \leftarrow B} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + C_{I \leftarrow B} [\vec{\omega}_{B/I,B}]_x \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{I \leftarrow E} [\vec{\omega}_{E/I,E}]_x \dot{\vec{x}}_{B/E,E} \\ &\quad + C_{I \leftarrow E} [\vec{\omega}_{E/I,E}]_x [\vec{\omega}_{E/I,E}]_x \vec{x}_{B/E,E} \end{aligned} \quad (9.184)$$

which can be transformed to ECEF frame coordinates as

$$\begin{aligned} \ddot{\vec{x}}_{B/I,E} &= C_{E \leftarrow B} \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + C_{E \leftarrow B} [\vec{\omega}_{B/I,B}]_x \begin{bmatrix} u \\ v \\ w \end{bmatrix} + [\vec{\omega}_{E/I,E}]_x C_{E \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\ &\quad + \begin{bmatrix} 0 & \omega_E & 0 \\ -\omega_E & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 & \omega_E & 0 \\ -\omega_E & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{x}_{B/E,E} \end{aligned} \quad (9.185)$$

which can be transformed to body-fixed frame coordinates as

$$\ddot{\vec{x}}_{B/I,B} = \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p + p_{E/I,B} \\ q + q_{E/I,B} \\ r + r_{E/I,B} \end{bmatrix}_x \begin{bmatrix} u \\ v \\ w \end{bmatrix} + [\vec{\omega}_{E/I,B}]_x \begin{bmatrix} u \\ v \\ w \end{bmatrix} + C_{B \leftarrow E} \begin{bmatrix} -\omega_E^2 & 0 & 0 \\ 0 & -\omega_E^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{x}_{B/E,E} \quad (9.186)$$

where

$$\vec{\omega}_{E/I,B} = \begin{bmatrix} p_{E/I,B} \\ q_{E/I,B} \\ r_{E/I,B} \end{bmatrix} \quad (9.187)$$

With this redefinition of the components of the inertial linear acceleration, one may then write the *translation equation of motion* as

$$\begin{bmatrix} X \\ Y \\ Z \end{bmatrix} + \vec{g}_B - \begin{bmatrix} q_{E/I,B}w - r_{E/I,B}v \\ r_{E/I,B}u - p_{E/I,B}w \\ p_{E/I,B}v - q_{E/I,B}u \end{bmatrix} - C_{B \leftarrow E} \begin{bmatrix} -\omega_E^2 & 0 & 0 \\ 0 & -\omega_E^2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{x}_{B/E,E} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (9.188)$$

where

$$\begin{bmatrix} p_{E/I,B} \\ q_{E/I,B} \\ r_{E/I,B} \end{bmatrix} = C_{B \leftarrow L}(\phi_L, \theta_L, \psi_L) \begin{bmatrix} \omega_E \cos \ell_L \\ 0 \\ -\omega_E \sin \ell_L \end{bmatrix} = C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} \omega_E \cos \ell \\ 0 \\ -\omega_E \sin \ell \end{bmatrix} \quad (9.189)$$

and

$$\vec{x}_{B/E,E} = \begin{bmatrix} d_E \cos \ell_L \cos \lambda \\ d_E \cos \ell_L \sin \lambda \\ d_E \sin \ell_L \end{bmatrix} = \begin{bmatrix} (N_\ell + h) \cos \lambda \cos \ell \\ (N_\ell + h) \sin \lambda \cos \ell \\ (N_\ell(1 - e_E^2) + h) \sin \ell \end{bmatrix} \quad (9.190)$$

Of particular note, if one assumes a **spherical-Earth model**, i.e. $\ell_L = \ell$ and $d_E = \bar{R}_E + h$ where \bar{R}_E is the **mean radius of the Earth**, defined as 6,366,707.0195 m by the WGS 84, one has the simpler equations

$$\begin{bmatrix} \dot{\ell} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{\bar{R}_E+h} & 0 & 0 \\ 0 & \frac{1}{(\bar{R}_E+h) \cos \ell} & 0 \\ 0 & 0 & -1 \end{bmatrix} C_{L \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (9.191)$$

and

$$\begin{bmatrix} X - g(\ell, h) \sin \theta \\ Y + g(\ell, h) \cos \theta \sin \phi \\ Z + g(\ell, h) \cos \theta \cos \phi \end{bmatrix} - \begin{bmatrix} q_{E/I,B}w - r_{E/I,B}v \\ r_{E/I,B}u - p_{E/I,B}w \\ p_{E/I,B}v - q_{E/I,B}u \end{bmatrix} - C_{B \leftarrow N} \begin{bmatrix} \omega_E^2 (\bar{R}_E + h) \cos \lambda \sin \lambda \\ 0 \\ \omega_E^2 (\bar{R}_E + h) \cos^2 \lambda \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \end{bmatrix} \quad (9.192)$$

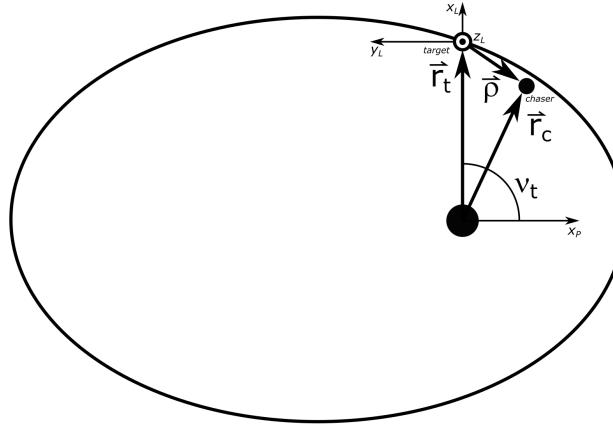
References

For more information, please refer to the following

- Schmidt, D. K., “2.6 Effects of Spherical, Rotating Earth,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 65-75
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “1.6 Geodesy, Coordinate Systems, Gravity,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 23-34
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “1.8 Advanced Topics,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 44-58

9.5 Relative Orbital Satellite Dynamics

In many instances for spacecraft dynamics and control, one is interested in modeling the dynamics between multiple spacecraft, e.g., a **rendezvous and proximity operation (RPO)**. To that end, consider the following simplified model of two satellites operating in proximity, i.e. a simplified **three-body problem**, involving a **chaser spacecraft** and a **target spacecraft** (subscript t) on an elliptical orbit. One can represent this relative motion in **Hill's frame (HF)** for the target spacecraft as shown in the following figure.



Thus, one can represent the target position relative to the celestial body in the target's HF axes as

$$\vec{r}_t = \begin{bmatrix} r_t \\ 0 \\ 0 \end{bmatrix} \quad (9.193)$$

and the relative position of the chaser in the target's HF axes as

$$\vec{\rho} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (9.194)$$

Then, noting

$$\vec{r}_c = \vec{r}_t + \vec{\rho} \quad (9.195)$$

and using Newton's Law of Gravitation for the relative motion of the chaser spacecraft relative to the target spacecraft, one has the equations of motion

$$\ddot{\vec{\rho}} + 2\vec{\omega}_{t,P/I} \times \dot{\vec{\rho}} + \vec{\alpha} \times \vec{\rho} + \vec{\omega}_{t,P/I} \times [\vec{\omega}_{t,P/I} \times \vec{\rho}] = \frac{\mu m_c}{\| \vec{r}_t + \vec{\rho} \|_2^3} (\vec{r}_t + \vec{\rho}) - \frac{\mu m_t}{r_t^3} \vec{r}_t \quad (9.196)$$

which is a nonlinear due to the gravity dependence on $\|r_c\|_2^{-3}$ and time-varying due to the \vec{r}_t dependence.

If one assumes a circular orbit for the target spacecraft and by linearizing about the resulting constant \vec{r}_t , one obtains the LTI Clohessy-Wiltshire (CW) equations for artificial satellites, also known as the Hill equations for natural satellites, in the target's HF as

$$\begin{aligned} \ddot{x} &= 2n_t \dot{y} + 3n_t^2 x + u_x \\ \ddot{y} &= -2n_t \dot{x} + u_y \\ \ddot{z} &= -n_t^2 z + u_z \end{aligned} \quad (9.197)$$

where $n = \sqrt{\mu/r_t^3}$ and $\vec{u} = [u_x \ u_y \ u_z]^T$ is the acceleration input, e.g., the mass-normalized thrust force, for the chaser satellite. Then, one has the continuous-time LTI dynamics equation

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \\ 3n_t^2 & 0 & 0 & 0 & 2n_t & 0 \\ 0 & 0 & 0 & -2n_t & 0 & 0 \\ 0 & 0 & -n_t^2 & 0 & 0 & 0 \end{bmatrix} \vec{x} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{u} \quad (9.198)$$

where

$$\vec{x} = \begin{bmatrix} \vec{\rho} \\ \dot{\vec{\rho}} \end{bmatrix} = \begin{bmatrix} x \\ y \\ z \\ \dot{x} \\ \dot{y} \\ \dot{z} \end{bmatrix} \quad (9.199)$$

This can also be discretized at Δt using a zero-order hold as the discrete-time LTI dynamics equation

$$\begin{aligned} \vec{x}[k+1] &= F\vec{x}[k] + G\vec{u}[k] \\ &= \begin{bmatrix} 4 - 3 \cos(n_t \Delta t) & 0 & 0 & n_t^{-1} \sin(n_t \Delta t) & 2n_t^{-1}(1 - \cos(n_t \Delta t)) & 0 \\ 6(\sin(n_t \Delta t) - n_t \Delta t) & 1 & 0 & -2n_t^{-1}(1 - \cos(n_t \Delta t)) & n_t^{-1}(4 \sin(n_t \Delta t) - n_t \Delta t) & 0 \\ 0 & 0 & \cos(n_t \Delta t) & 0 & 0 & n_t^{-1} \sin(n_t \Delta t) \\ 3n_t \sin(n_t \Delta t) & 0 & 0 & \cos(n_t \Delta t) & 2 \sin(n_t \Delta t) & 0 \\ -6n_t(1 - \cos(n_t \Delta t)) & 0 & 0 & -2 \sin(n_t \Delta t) & 4 \cos(n_t \Delta t) - 3 & 0 \\ 0 & 0 & -n_t \sin(n_t \Delta t) & 0 & 0 & \cos(n_t \Delta t) \end{bmatrix} \vec{x} \\ &\quad + \begin{bmatrix} n_t^{-2}(1 - \cos(n_t \Delta t)) & 2n_t^{-2}(n_t \Delta t - \sin(n_t \Delta t)) & 0 \\ -2n_t^{-2}(n_t \Delta t - \sin(n_t \Delta t)) & 4n_t^{-2}(1 - \cos(n_t \Delta t)) - \frac{3}{2}n_t \Delta t^2 & 0 \\ 0 & 0 & n_t^{-2}(1 - \cos(n_t \Delta t)) \\ n_t^{-1} \sin(n_t \Delta t) & 2n_t^{-1}(1 - \cos(n_t \Delta t)) & 0 \\ -2n_t^{-1}(1 - \cos(n_t \Delta t)) & 4n_t^{-1} \sin(n_t \Delta t) - 3\Delta t & 0 \\ 0 & 0 & n_t^{-1} \sin(n_t \Delta t) \end{bmatrix} \vec{u}[k] \end{aligned} \quad (9.200)$$

References

For more information, please refer to the following

- Jewison, C., "Guidance and Control for Multi-Stage Rendezvous and Docking Operations in the Presence of Uncertainty," *Ph.D. Thesis*, Massachusetts Inst. of Technology, 2017

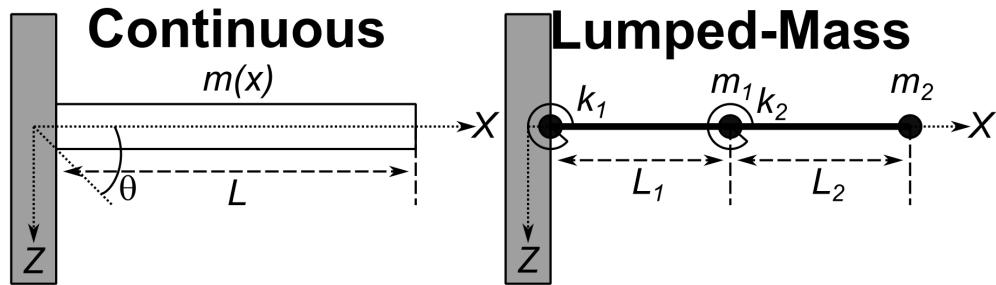
Elastic Aerospace Vehicle Dynamics

10.1 Elastic-Body Dynamics

As noted previously, introductory FDC assumes that rigid-body dynamics can be used to model the equations of motion for flight vehicles. However, in reality, flight vehicle structures are not rigid, but have some elasticity or flexibility in the structure. Thus, this chapter will look at the additional modeling of structural vibrations in the flight vehicle equations of motion, highlighting elastic airplane dynamics.

Fixed-Beam Vibration Problem

To contextualize this topic, consider the following two figures for a fixed-beam vibration problem



In the left figure of the fixed-beam, one has a continuous deformable body with some mass distribution, $m(x)$, as a function of the horizontal coordinate, x . It can be shown that the partial differential equation (PDE) governing the vertical deformation of the beam, Z , is given by

$$\frac{\partial^2}{\partial x^2} \left(EI(x) \frac{\partial^2 Z(x, t)}{\partial x^2} \right) + m(x) \frac{\partial^2 Z(x, t)}{\partial t^2} = 0 \quad (10.1)$$

where E is the elastic modulus of the beam material and I is the area moment of inertia of the beam cross-section about its neutral axis. Using the separation of variables technique, one can write the solution to this PDE as

$$Z(x, t) = \sum_{i=1}^{\infty} v_i(x) \eta_i(t) \quad (10.2)$$

which is an infinite sum of terms, each consisting of a purely space-dependent function $v_i(x)$ and a purely time-dependent function, $\eta_i(t)$. The functions $v_i(x)$ are called the **mode shapes**, also known as the **eigenfunctions**, and the functions $\eta_i(t)$ are called the **modal coordinates**. As the solution is an infinite sum, the beam-vibration problem is called an **infinite-dimensional problem**.

To obtain a finite-dimensional approximation to this infinite-dimensional problem, one may simplify the continuous beam model as a discrete mass model with $i = 1, \dots, n$ particles, a.k.a. **lumped-mass model**, where each mass particle, m_i , has an associated spring stiffness, k_i , and different particles are connected by massless rigid rods of lengths L_i . It should be noted that the solution to a vibration problem using a finite-element analysis (FEA) will result in a lumped-mass approximation. Thus, lumped-mass approximations are always used for real, complex, flight vehicle structures in FDC. As such, in the right figure of the beam-vibration example, the continuous beam has been approximated by two masses, two springs, and two rods. Note that if one desired to have a better approximation, one would simply add more lumped-masses to the beam.

In Lagrangian mechanics, one can solve for the equations of motion for these lumped-mass systems using the Lagrangian $L = T - U$ where T is the kinetic energy of the system and U is the potential (or strain) energy of the system. Then, with no external forces acting on the system, the calculus of variations provides the **Euler-Lagrange equation of motion**

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{q}_i} \right) - \frac{\partial T}{\partial q_i} + \frac{\partial U}{\partial q_i} = 0 \quad (10.3)$$

where q_i is the i -th generalized coordinate used to describe the system. These **generalized coordinates** can include both physical and nonphysical coordinates and will be discussed next lecture in detail. Note that this course will make continually use of the Euler-Lagrange equation in the treatment of elastic-body dynamics and will be invoked without proof as it's derivation and properties are beyond the scope of this course and typically introduced in graduate level dynamics course.

For the simple beam-vibration example previously, possible coordinates to describe the beam's motion could be the transverse displacements of the masses $Z_i(t)$, or the angular displacements of the masses, $\theta_i(t)$ which are related by

$$\begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ L_1 + L_2 & L_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (10.4)$$

Recalling systems-of-particles dynamics, the kinetic energy of the beam can be written as

$$T = \frac{1}{2} (m_1 \dot{Z}_1^2 + m_2 \dot{Z}_2^2) = \frac{1}{2} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{bmatrix}^T \begin{bmatrix} m_1 & 0 \\ 0 & m_2 \end{bmatrix} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \end{bmatrix} \quad (10.5)$$

and the potential energy of the beam can be written as

$$U = \frac{1}{2} (k_1 \theta_1^2 + k_2 \theta_2^2) = \frac{1}{2} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix}^T \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (10.6)$$

Then, using Lagrange's equation, one has

$$\begin{bmatrix} m_1^2 L_1^2 + m_2(L_1 + L_2)^2 & m_1 L_2(L_1 + L_2) \\ m_1 L_2(L_1 + L_2) & m_2 L_2^2 \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (10.7)$$

which demonstrates the general matrix-vector differential form for *all* lumped-mass vibration problems

$$M \ddot{\vec{q}} + K \vec{q} = \vec{0} \quad (10.8)$$

where M is the **mass matrix** and K is the **stiffness matrix**, both of which will be real, symmetric matrices and M will always be positive-definite, i.e. have strictly positive eigenvalues. Thus, a vibration problem can be completely described by selecting the generalized coordinates, setting up the initial conditions, and finding the mass and stiffness matrices.

Modal Analysis of Lumped-Mass Vibration

Furthermore, this general lumped-mass vibration problem can be rewritten as

$$\ddot{\vec{q}} + D \vec{q} = \vec{0} \quad (10.9)$$

where $D = M^{-1}K$ is the **dynamic matrix** and always exists since M is positive-definite. Then, using the standard transformation

$$\vec{q} = \Psi \vec{\eta} \quad (10.10)$$

where Ψ is the **modal matrix** of D consisting of its n eigenvectors, \vec{v}_i where $i = 1, \dots, n$. Furthermore, one can write

$$\Psi^{-1} D \Psi = \Lambda \quad (10.11)$$

where Λ is a diagonal matrix of the n eigenvalues of D , λ_i where $i = 1, \dots, n$.

Recall that the eigenvectors satisfy the equation

$$(\lambda_i I - D) \vec{v}_i = \vec{0} \quad (10.12)$$

and

$$\Psi = [\vec{v}_1 | \vec{v}_2 | \cdots | \vec{v}_n] \quad (10.13)$$

Thus, the general lumped-mass vibration problem becomes

$$\ddot{\vec{\eta}} + \Lambda \vec{\eta} = \vec{0} \quad (10.14)$$

whose n differential equations are now independent, i.e.

$$\ddot{\eta}_i + \lambda_i \eta_i = 0 \quad (10.15)$$

for $i = 1, \dots, n$ where each has a solution

$$\eta_i(t) = A_i \cos(\sqrt{\lambda_i} t + \Gamma_i) \quad (10.16)$$

where the constants of integration, A_i and Γ_i , depend on initial conditions. Thus, for general lumped-mass vibration problems, one can find n natural modes each oscillating at natural frequencies $\omega_i = \sqrt{\lambda_i}$. Furthermore, from the definition of Ψ , one has

$$\vec{q}(t) = [\vec{v}_1 | \vec{v}_2 | \cdots | \vec{v}_n] \vec{\eta}(t) = \sum_{i=1}^n \vec{v}_i \eta_i(t) \quad (10.17)$$

where each modal response η_i contributes to the system response through the eigenvectors or mode shapes. As eigenvectors have arbitrary magnitude, they are typically normalized to a unit length, unity displacement of a selected element, or unity generalized mass as will be shown.

Recall that by definition of D and \vec{v}_i , one has

$$(\lambda_i I - M^{-1} K) \vec{v}_i = 0 \quad (10.18)$$

or

$$\lambda_i M \vec{v}_i = K \vec{v}_i \quad (10.19)$$

Thus, if one multiplies by another eigenvector such that

$$\lambda_i \vec{v}_j^T M \vec{v}_i = v_j^T K \vec{v}_i \quad (10.20)$$

and by the same process, one also has

$$\lambda_j \vec{v}_i^T M \vec{v}_j = v_i^T K \vec{v}_j \quad (10.21)$$

Finally by noting that M and K are symmetric, one can further write that

$$(\lambda_i - \lambda_j) \vec{v}_j^T M \vec{v}_i = 0 \quad (10.22)$$

and

$$(\lambda_i - \lambda_j) \vec{v}_j^T K \vec{v}_i = 0 \quad (10.23)$$

which means that if $\lambda_i \neq \lambda_j \forall i \neq j$, then the **orthogonality property** holds for the restrained lumped-mass modes, i.e.

$$\vec{v}_j^T M \vec{v}_i = 0, i \neq j \quad (10.24)$$

and

$$\vec{v}_j^T K \vec{v}_i = 0, i \neq j \quad (10.25)$$

Furthermore if $i = j$, one then can define the **i-th generalized mass**

$$\mathcal{M}_i = \vec{v}_i^T M \vec{v}_i \quad (10.26)$$

and **i-th generalized stiffness**

$$\mathcal{K}_i = \vec{v}_i^T K \vec{v}_i \quad (10.27)$$

Finally, by this orthogonality property, the definition of Ψ , and defining $\mathcal{M} = \text{diag}[\mathcal{M}_1, \dots, \mathcal{M}_n]$ and $\mathcal{K} = \text{diag}[\mathcal{K}_1, \dots, \mathcal{K}_n]$, one can alternatively write the lumped-mass vibration equations of motion as

$$\Psi^{-1} M \Psi \ddot{\vec{\eta}} + \Psi^{-1} K \Psi \vec{\eta} = \vec{0} \quad (10.28)$$

$$\mathcal{M}\ddot{\vec{\eta}} + \mathcal{K}\vec{\eta} = \vec{0} \quad (10.29)$$

or

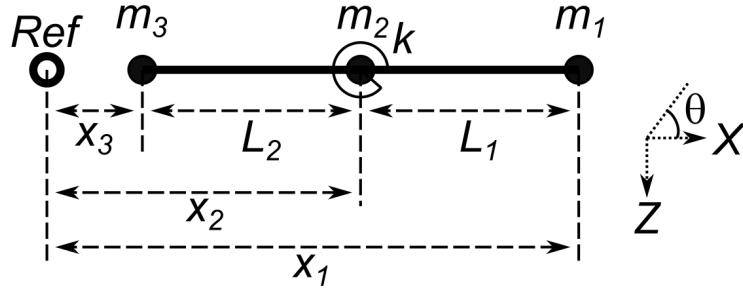
$$\ddot{\vec{\eta}} + \mathcal{M}^{-1}\mathcal{K}\vec{\eta} = \vec{0} \quad (10.30)$$

which demonstrates that

$$\lambda_i = \frac{\mathcal{M}_i}{\mathcal{K}_i} \quad (10.31)$$

Unrestrained Lumped-Mass Model

Consider the following unrestrained three-lumped-mass system which is free to translate and rotate



where first only the vertical displacement Z will be analyzed. Note that the bending displacement of the beam occurs by the relative deflection angle θ between the lines for rods 1 and 2. This deflection angle can be shown to be (for small angle approximation)

$$\theta = \frac{Z_1 - Z_2}{x_1 - x_2} - \frac{Z_2 - Z_3}{x_2 - x_3} = \left[\frac{1}{x_1 - x_2} \quad -\frac{1}{x_1 - x_2} - \frac{1}{x_2 - x_3} \quad \frac{1}{x_2 - x_3} \right] \vec{Z} = C \vec{Z} \quad (10.32)$$

where C is a constraint matrix that relates the beam-displacement coordinates.

Similar to the restrained beam, the kinetic energy of the beam can be written as

$$T = \frac{1}{2} \left(m_1 \dot{Z}_1^2 + m_2 \dot{Z}_2^2 + m_3 \dot{Z}_3^2 \right) = \frac{1}{2} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \end{bmatrix}^T \begin{bmatrix} m_1 & 0 & 0 \\ 0 & m_2 & 0 \\ 0 & 0 & m_3 \end{bmatrix} \begin{bmatrix} \dot{Z}_1 \\ \dot{Z}_2 \\ \dot{Z}_3 \end{bmatrix} = \frac{1}{2} \dot{\vec{Z}}^T M \dot{\vec{Z}} \quad (10.33)$$

and the potential energy of the beam can be written as

$$U = \frac{1}{2} k \theta^2 = \frac{1}{2} \theta^T k \theta = \frac{1}{2} \vec{Z}^T C^T k C \vec{Z} = \frac{1}{2} \vec{Z}^T K_c \vec{Z} \quad (10.34)$$

where K_c is the **constrained stiffness matrix**. Then, again using Euler-Lagrange's equation, one has

$$\mathcal{M}\ddot{\vec{Z}} + K_c \vec{Z} = \vec{0} \quad (10.35)$$

or defining $D_c = M^{-1}K_c$ as the **constrained dynamic matrix**, one has

$$\ddot{\vec{Z}} + D_c \dot{\vec{Z}} = \vec{0} \quad (10.36)$$

Continuing with the modal analysis one has

$$\lambda_i \vec{v}_i = D_c \vec{v}_i \quad (10.37)$$

or for two eigenvalues/eigenvectors

$$\lambda_i M \vec{v}_i = K_c \vec{v}_i \quad (10.38)$$

$$\lambda_j M \vec{v}_j = K_c \vec{v}_j \quad (10.39)$$

and repeating the previous equations as M and K_c are still symmetric, one has

$$(\lambda_i - \lambda_j) \vec{v}_j^T M \vec{v}_i = 0 \quad (10.40)$$

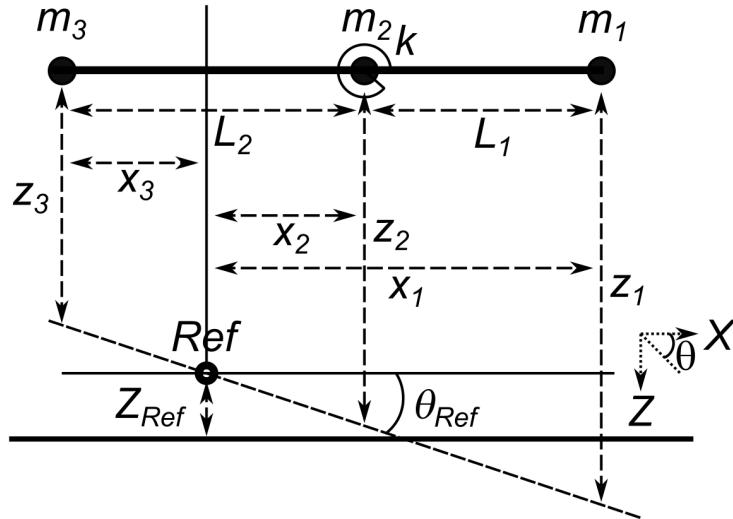
and

$$(\lambda_i - \lambda_j) \vec{v}_i^T K_c \vec{v}_i = 0 \quad (10.41)$$

However, for the unrestrained beam, two of the eigenvalues of D can be shown to be 0 and hence equal due to the existence of two rigid-body degrees-of-freedom (DOF), vertical translation and rotation of the *entire* beam. Thus, this system will have two rigid-body modes and a single vibration mode corresponding to the non-zero eigenvalue and associated eigenvector.

Thus, further work must be done to describe the entire elastic-body motion in terms of mutually orthogonal or **normal** modes. From linear algebra, if a matrix has repeated eigenvalues, *any* linear combination of the eigenvectors associated with the repeated eigenvalues are also eigenvectors of the given matrix. This fact can be used to obtain mutually orthogonal modes for unrestrained bodies.

Before this is derived further, consider an alternate approach that will consider the rigid-body degrees-of-freedom more directly, in particular consider the total, i.e., inertial, vertical position of the masses are referenced to some arbitrary reference axis, as demonstrated in the following three-lumped-mass example



where Ref is the arbitrary reference point for the positional coordinates. Here the total vertical displacements of the lumped-masses (with the small angle approximation) are

$$\begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} = \begin{bmatrix} Z_{Ref} \\ Z_{Ref} \\ Z_{Ref} \end{bmatrix} + \begin{bmatrix} x_1 \\ x_2 \\ -x_3 \end{bmatrix} \theta_{Ref} + \begin{bmatrix} z_1 \\ z_2 \\ z_3 \end{bmatrix} \quad \vec{Z} = \vec{1} Z_{Ref} + \vec{x} \theta_{Ref} + \vec{z} \quad (10.42)$$

Using Lagrange's equation for the equations of motion, the kinetic energy can be written as

$$T = \frac{1}{2} \vec{Z}^T M \vec{\dot{Z}} \quad (10.43)$$

and the potential (or strain) energy can be written as

$$U = \frac{1}{2} \theta^T k \theta \quad (10.44)$$

As before, one can model the angular displacement as

$$\theta = \frac{z_1 - z_2}{x_1 - x_2} - \frac{z_2 - z_3}{x_2 + x_3} = \begin{bmatrix} \frac{1}{x_1 - x_2} & \left(-\frac{1}{x_1 - x_2} - \frac{1}{x_2 + x_3} \right) & \frac{1}{x_2 + x_3} \end{bmatrix} \vec{z} = C \vec{z} \quad (10.45)$$

Thus,

$$U = \frac{1}{2} \vec{z}^T C^T k C \vec{z} = \frac{1}{2} \vec{z}^T K_c \vec{z} \quad (10.46)$$

where it should be noted that T is a function of the inertial velocities, while U is a function of the relative displacements. This will need to be resolved before using Lagrange's equation on the generalized coordinates.

Next, two additional constraints must be imposed. First, in the absence of external forces and moments, both translational and rotational momenta must be constant. Taking this arbitrary constant to 0 results in

$$m_1 \dot{Z}_1 + m_2 \dot{Z}_2 + m_3 \dot{Z}_3 = \vec{1}^T M \vec{\dot{Z}} = 0 \quad (10.47)$$

and

$$m_1 x_1 \dot{Z}_1 + m_2 x_2 \dot{Z}_2 - m_3 x_3 \dot{Z}_3 = \vec{x}^T M \vec{\dot{Z}} = 0 \quad (10.48)$$

These two constraints imply that $\vec{\dot{Z}}$ must be orthogonal (with respect to M) to the vectors $\vec{1}$ and \vec{x} . Therefore, one can define \vec{Z}_c and \vec{z} as the absolute and relative vertical displacement velocity which satisfy these constraints, respectively. Furthermore, from this analysis one can infer that $\vec{1}$ and \vec{X} may be appropriate rigid-body mode shapes. If so, then these two shapes must be mutually orthogonal (with respect to M), i.e.

$$m_1 x_1 + m_2 x_2 - m_3 x_3 = \vec{1}^T M \vec{x} = 0 \quad (10.49)$$

which is equivalent to setting the reference point, Ref , at the center of mass of the beam, point G . Lastly, recall that the total mass of the beam can be written as

$$M_{tot} = \vec{1}^T M \vec{1} \quad (10.50)$$

and the moment of inertia of the beam about its center of mass, G , can be written as

$$I_G = \sum_{i=1}^3 m_i x_i^2 = \vec{x}^T M \vec{x} \quad (10.51)$$

Then, rewriting Equation 10.42 in terms of constrained displacements, one has

$$\vec{Z}_c = \vec{1} Z_{Ref} + \vec{x} \theta_{Ref} + \vec{z}_c \quad (10.52)$$

from which if one can invoke the constraints and relative motion \vec{z}_c in terms of mutually orthogonal modal responses, then the desired solution to the vibration problem will be derived.

To that end, differentiating Equation 10.52 with respect to time and using the momenta constraints, one has

$$\vec{1}^T M \dot{\vec{Z}}_c = \vec{1}^T M \left[\vec{1} \dot{Z}_{Ref} + \vec{x} \dot{\theta}_{Ref} + \dot{\vec{z}}_c \right] = 0 \quad (10.53)$$

and

$$\vec{x}^T M \dot{\vec{Z}}_c = \vec{x}^T M \left[\vec{1} \dot{Z}_{Ref} + \vec{x} \dot{\theta}_{Ref} + \dot{\vec{z}}_c \right] = 0 \quad (10.54)$$

Then, noting the total mass and moment of inertia equations and center of mass constraint, one has

$$\dot{Z}_{Ref} = -\frac{1}{M_{tot}} \vec{1}^T M \dot{\vec{z}}_c \quad (10.55)$$

and

$$\dot{\theta}_{Ref} = \frac{1}{I_G} \vec{x}^T M \dot{\vec{z}}_c \quad (10.56)$$

Finally, applying these two constraints, one can write that constrained total velocities as functions of the constrained relative velocities as

$$\dot{\vec{Z}}_c = \left[I_{3 \times 3} - \frac{1}{M_{tot}} \vec{1} \vec{1}^T M - \frac{1}{I_G} \vec{x} \vec{x}^T M \right] \dot{\vec{z}}_c = \Xi \dot{\vec{z}}_c \quad (10.57)$$

Then, the kinetic energy in terms of the constrained relative velocities can be written as

$$T = \frac{1}{2} \dot{\vec{z}}_c^T \Xi^T M \Xi \dot{\vec{z}}_c = \frac{1}{2} \dot{\vec{z}}_c^T M_c \dot{\vec{z}}_c \quad (10.58)$$

and the potential (or strain) energy in terms of the constrained relative displacements can be written as

$$U = \frac{1}{2} \vec{z}_c^T K_c \vec{z}_c \quad (10.59)$$

Then utilizing \vec{z}_c as the generalized coordinates \vec{q} in Lagrange's equation in vector form, i.e.

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\vec{q}}} \right) - \frac{\partial T}{\partial \vec{q}} + \frac{\partial U}{\partial \vec{q}} = \vec{0}^T \quad (10.60)$$

one has for the unrestrained beam-vibration equations of motion

$$M_c \ddot{\vec{z}}_c + K_c \vec{z}_c = \vec{0} \quad (10.61)$$

where the constrained mass matrix M_c is now singular and its inverse does not exist. Thus, to solve for the mode shapes and vibration frequencies, one must solve the **generalized eigenvalue problem**, i.e.

$$(\lambda_i M_c - K_c) \vec{v}_i = 0 \quad (10.62)$$

Modal Analysis of Generalized Eigenvalue Problem

With the unrestrained beam, the modal analysis of the generalized eigenvalue problem will provide a model for incorporating vibration modes into the rigid-body dynamics to produce elastic-body dynamics. Recall the generalized eigenvalue problem from the previous section

$$\lambda_i M_c \vec{v}_i = K_c \vec{v}_i \quad (10.63)$$

So, as before, consider two i, j pairs of generalized eigenvalues and eigenvectors

$$(\lambda_i - \lambda_j) \vec{v}_i^T M_c \vec{v}_j = 0 \quad (10.64)$$

However, when $i \neq j$ and the eigenvalues are distinct, e.g. not both zero, this equation assures that the two associated eigenvectors are orthogonal with respect to the *constrained* mass matrix, not the mass matrix M , as required. From the definition of M_c , one can write

$$(\lambda_i - \lambda_j) \vec{v}_i^T \Xi^T M \Xi \vec{v}_j = 0 \quad (10.65)$$

or defining the **constrained eigenvectors** as

$$\vec{v}_{c,i} = \Xi \vec{v}_i \quad (10.66)$$

Thus, when $i \neq j$ and the two eigenvalues are distinct, $\vec{v}_{c,i}$ are mutually orthogonal with respect to the mass matrix M . Note that for the unrestrained beam example, two of the transformed generalized eigenvectors will be equal to $\vec{0}$ due to the two rigid-body modes and one will satisfy the three orthogonal constraints, the single vibration mode shape \vec{v}_{vib} . For an n -lumped-mass beam, one can extend this to the relative displacement equation for $n - 2$ vibration modes

$$\vec{z}_c(t) = \sum_{i=1}^{n-2} \vec{v}_{vib,i} A_i \cos(\omega_{vib,i} t + \Gamma_i) \quad (10.67)$$

Note that all *vibration* mode shapes will be mutually orthogonal with respect to the mass matrix. To prove that these are also orthogonal to the rigid-body mode shapes, \vec{l} and \vec{x} , first consider the relative motion \vec{z}_c as a function of the original eigenvectors.

$$\vec{z}_c(t) = \sum_{i=1}^n \vec{v}_i \eta_i(t) \quad (10.68)$$

Next, recall the orthogonality constraints for the $\dot{\vec{Z}}_c$ relative to \vec{l} and \vec{x} , i.e.

$$\vec{l}^T M \dot{\vec{Z}}_c = 0 \quad (10.69)$$

and

$$\vec{x}^T M \dot{\vec{Z}}_c = 0 \quad (10.70)$$

as well as the relation

$$\dot{\vec{Z}}_c = \Xi \dot{\vec{z}}_c \quad (10.71)$$

Thus, one has

$$\vec{1}^T M \dot{\vec{Z}}_c = \vec{1}^T M \Xi \dot{\vec{z}}_c = \vec{1}^T M \Xi \sum_{i=1}^n \vec{v}_{c,i} \dot{\eta}_i(t) = 0 \quad (10.72)$$

and

$$\vec{x}^T M \dot{\vec{Z}}_c = \vec{x}^T M \Xi \dot{\vec{z}}_c = \vec{x}^T M \Xi \sum_{i=1}^n \vec{v}_{c,i} \dot{\eta}_i(t) = 0 \quad (10.73)$$

Finally, by inspection, this requires

$$\vec{1}^T M \Xi \vec{v}_i = \vec{1}^T M \vec{v}_{c,i} = 0 \quad \forall i \quad (10.74)$$

and

$$\vec{x}^T M \Xi \vec{v}_i = \vec{x}^T M \vec{v}_{c,i} = 0 \quad \forall i \quad (10.75)$$

These equations signify that all the constrained eigenvectors are orthogonal to \vec{x} with respect to M . Thus, these constrained eigenvectors must be the desired orthogonal vibration mode shapes. Furthermore, all these eigenvectors (including $\vec{1}$ and \vec{x}) are mutually orthogonal with respect to M .

The key idea for orthogonality among the modal decomposition is that the physical responses of the unrestrained beam can be expressed in terms of the linear combination of *mutually orthogonal* modes, i.e.

$$\sum_{i=1}^n \vec{v}_{c,i} \eta_i = \vec{1} \eta_n + \vec{x} \eta_{n-1} + \sum_{i=1}^{n-2} \vec{v}_{vib,i} \eta_i \quad (10.76)$$

where the rigid-body and vibration modes derived previously are also known as the **normal modes**. Then, the unrestrained three-lumped mass model had the following model for the vertical displacements (with the small angle approximation)

$$\begin{aligned} \vec{Z}(t) &= \vec{1} Z_{Ref}(t) + \vec{x} \theta_{Ref}(t) + \vec{z}_{vib}(t) \\ \vec{Z}(t) &= \vec{1} Z_{Ref}(t) + \vec{x} \theta_{Ref}(t) + \sum_{i=1}^{n-2} \vec{v}_{vib,i} \eta_i(t) \end{aligned} \quad (10.77)$$

Thus, with this consideration, one can form generalized coordinates for deriving the equation of motion for the vertical displacement of an elastic-body as

$$\begin{aligned} \vec{Z} &= [\vec{1} \quad \vec{x} \quad \vec{v}_{vib,1} \quad \cdots \quad \vec{v}_{vib,n-2}] \begin{bmatrix} Z_{Ref} \\ \theta_{Ref} \\ \eta_1 \\ \vdots \\ \eta_{n-2} \end{bmatrix} \\ &= \Psi \vec{q} \end{aligned} \quad (10.78)$$

Then, the kinetic energy of the beam can be rewritten

$$\begin{aligned} T &= \frac{1}{2} \dot{\vec{Z}}^T M \dot{\vec{Z}} = \frac{1}{2} \dot{\vec{q}}^T \Psi^T M \Psi \dot{\vec{q}} \\ &= \frac{1}{2} M_{tot} \dot{Z}_{Ref}^2 + \frac{1}{2} I_G \dot{\theta}_{Ref}^2 + \frac{1}{2} \dot{\vec{\eta}}_{vib}^T \mathcal{M}_{vib} \dot{\vec{\eta}}_{vib} \\ &= \frac{1}{2} \dot{\vec{q}}^T \mathcal{M} \dot{\vec{q}} \end{aligned} \quad (10.79)$$

where the rigid-body kinetic energy terms and elastic kinetic energy terms notably linearly combine, the **generalized mass matrix** is

$$\mathcal{M} = \begin{bmatrix} M_{tot} & 0 & 0 \\ 0 & I_G & 0 \\ 0 & 0 & \mathcal{M}_{vib} \end{bmatrix}, \quad (10.80)$$

and

$$\vec{\eta}_{vib} = [\eta_1 \ \cdots \ \eta_{n-2}]^T. \quad (10.81)$$

The potential (or strain) energy can be rewritten as

$$U = \frac{1}{2} \vec{z}^T K_c \vec{z} = \frac{1}{2} \vec{\eta}_{vib}^T \Psi_{vib}^T K_c \Psi_{vib} \vec{\eta}_{vib} = \frac{1}{2} \vec{\eta}_{vib}^T \mathcal{K}_{vib} \vec{\eta}_{vib} \quad (10.82)$$

where \mathcal{K}_{vib} is the generalized stiffness matrix and Ψ_{vib} is the vibration modal matrix, i.e.

$$\Psi_{vib} = [\vec{v}_{vib,1} \ \cdots \ \vec{v}_{vib,n-2}] \quad (10.83)$$

Finally, using Lagrange's equation for these energy expressions, one has

$$\mathcal{M} \ddot{\vec{q}} + \mathcal{K} \vec{q} = \begin{bmatrix} M_{tot} & 0 & 0 \\ 0 & I_G & 0 \\ 0 & 0 & \mathcal{M}_{vib} \end{bmatrix} \ddot{\vec{q}} + \begin{bmatrix} 0 & 0 \\ 0 & \mathcal{K}_{vib} \end{bmatrix} \vec{q} = 0 \quad (10.84)$$

which can alternatively be written as

$$\begin{aligned} M_{tot} \ddot{Z}_{Ref} &= 0 \\ I_G \ddot{\theta}_{Ref} &= 0 \\ \mathcal{M}_{vib} \ddot{\vec{\eta}}_{vib} + \mathcal{K}_{vib} \vec{\eta}_{vib} &= 0 \end{aligned} \quad (10.85)$$

for the free response of the unrestrained beam's EOMs which will be fundamental for the elastic-body flight dynamics.

Forced Motion and Virtual Work

Next, consider the addition of external forces on the lumped-mass systems which will result in how one can incorporate the aerodynamic forces into the elastic-body flight dynamics. To do so, consider Lagrange's equation with external forces applied

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\vec{q}}} \right) - \frac{\partial T}{\partial \vec{q}} + \frac{\partial U}{\partial \vec{q}} = \vec{Q} \quad (10.86)$$

where Q is the generalized force, i.e.

$$\vec{Q}^T = \frac{\partial \delta W}{\partial \delta \vec{q}} \quad (10.87)$$

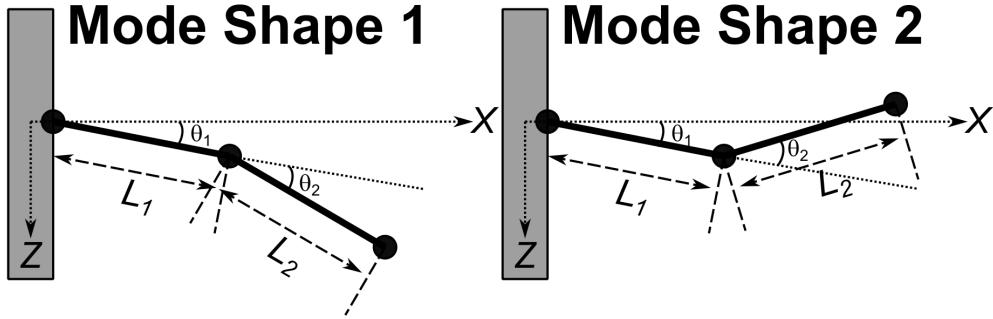
where $\delta \vec{q}$ is a virtual displacement of the generalized coordinates \vec{q} and the **virtual work** is defined as

$$\delta W = \sum_{i=1}^m \vec{F}_i \cdot \delta \vec{d}_i$$

$$\delta W = [\vec{F}_1^T \quad \dots \quad \vec{F}_m^T]^T \begin{bmatrix} \delta \vec{d}_1 \\ \vdots \\ \delta \vec{d}_m \end{bmatrix} \quad (10.88)$$

where $\delta \vec{d}_i$ is the virtual displacement of the point of application of force F_i .

As an example, consider the vertical displacement of two-lumped-mass restrained beam shown here with its two mode shapes.



Defining \hat{k} as the unit vector for the Z direction, one can define the vertical forces as

$$\vec{F}_1 = F_1 \hat{k} \quad \vec{F}_2 = F_2 \hat{k} \quad (10.89)$$

and the virtual physical displacements at the points of application as

$$\delta \vec{d}_1 = \delta Z_1 \hat{k}$$

$$\delta \vec{d}_2 = \delta Z_2 \hat{k} \quad (10.90)$$

Thus,

$$\delta W = F_1 \delta Z_1 + F_2 \delta Z_2$$

$$\delta W = [F_1 \quad F_2] \begin{bmatrix} \delta Z_1 \\ \delta Z_2 \end{bmatrix} \quad (10.91)$$

where

$$\begin{bmatrix} \delta Z_1 \\ \delta Z_2 \end{bmatrix} = \begin{bmatrix} L_1 & 0 \\ L_1 + L_2 & L_2 \end{bmatrix} \begin{bmatrix} \delta \theta_1 \\ \delta \theta_2 \end{bmatrix} \quad (10.92)$$

Now, assuming that the unforced (i.e. “free”) vibration problem has been solved in terms of θ_1 and θ_2 , then the elements of the free-vibration mode shapes would correspond to angular displacements and the

virtual displacements $\delta\theta_1$ and $\delta\theta_2$ can be expressed in terms of these mode shapes and two vibration modal coordinates, η_1 and η_2 as

$$\begin{bmatrix} \delta\theta_1 \\ \delta\theta_2 \end{bmatrix} = [\vec{v}_1 \quad \vec{v}_2] \begin{bmatrix} \delta\eta_1 \\ \delta\eta_2 \end{bmatrix}$$

$$\delta\vec{\theta} = \Psi\delta\vec{\eta} \quad (10.93)$$

Then, by substitution the virtual work can be written as

$$\delta W = [F_1 \quad F_2] \begin{bmatrix} L_1 & 0 \\ L_1 + L_2 & L_2 \end{bmatrix} \Psi\delta\vec{\eta}$$

$$\delta W = \vec{\mathcal{F}}^T \Psi\delta\vec{\eta} \quad (10.94)$$

where $\vec{\mathcal{F}}$ is the vector of applied forces relative to the virtual displacements. Finally, recall one can write the kinetic energy in terms of the modal coordinates as

$$T = \frac{1}{2} \dot{\vec{\theta}}^T M \dot{\vec{\theta}} = \frac{1}{2} \dot{\vec{\eta}}^T \Psi^T M \Psi \dot{\vec{\eta}} = \frac{1}{2} \dot{\vec{\eta}}^T \mathcal{M} \dot{\vec{\eta}} \quad (10.95)$$

and the potential (or strain) energy in terms of the modal coordinates as

$$U = \frac{1}{2} \vec{\theta}^T K \vec{\theta} = \frac{1}{2} \vec{\eta}^T \Psi^T K \Psi \vec{\eta} = \frac{1}{2} \vec{\eta}^T \mathcal{K} \vec{\eta} \quad (10.96)$$

Then, using Lagrange's equation with external forces applied results in

$$\mathcal{M}\ddot{\vec{\eta}} + \mathcal{K}\vec{\eta} = Q = \Psi^T \vec{\mathcal{F}} \quad (10.97)$$

Furthermore, for the unrestrained three-lumped-mass beam as described earlier, one can derive a similar expression which results in the form

$$M_{tot} \ddot{\vec{Z}}_{Ref} = F_1 + F_2 + F_3 = \vec{1}^T \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix}$$

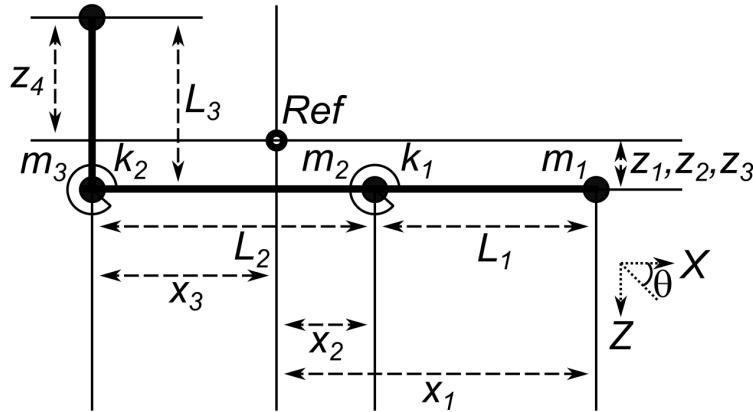
$$I_G \ddot{\theta}_{Ref} = F_1 x_1 + F_2 x_2 - F_3 x_3 = \vec{x}^T \begin{bmatrix} F_1 \\ F_2 \\ F_3 \end{bmatrix} \quad (10.98)$$

$$\mathcal{M}_{vib} \ddot{\vec{\eta}}_{vib} + \mathcal{K}_{vib} \vec{\eta}_{vib} = Q = \Psi_{vib}^T \vec{\mathcal{F}} = \vec{v}_{vib}^T \vec{\mathcal{F}}$$

for the forced response of the unrestrained beam's EOMs which will be fundamental for the elastic-body flight dynamics. Recall that the *Ref* here is the center of mass of the lumped-mass system. This can easily be extended to n -lumped-mass systems.

Multi-Directional Elastic Motion

Lastly, to extend these results to multi-directional motion requires more generalized vectors and matrices as each element of a mode shape/eigenvector can only correspond to direction of motion. To demonstrate this, consider the following bi-directional example of an unforced 2D truss.



where the directions of motion are X and Z and the reference point, Ref is the center of mass of the truss. The kinetic energy of the truss can be written as

$$\begin{aligned} T &= \frac{1}{2} [m_1(\dot{X}_1^2 + \dot{Z}_1^2) + m_2(\dot{X}_2^2 + \dot{Z}_2^2) + m_3(\dot{X}_3^2 + \dot{Z}_3^2) + m_4(\dot{X}_4^2 + \dot{Z}_4^2)] \\ &= \frac{1}{2} [\dot{\vec{X}}^T \quad \dot{\vec{Z}}^T] \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} \dot{\vec{X}} \\ \dot{\vec{Z}} \end{bmatrix} \end{aligned} \quad (10.99)$$

The potential (or strain) energy of the truss can be written as

$$U = \frac{1}{2} k_1 \theta_1^2 + \frac{1}{2} k_2 \theta_2^2 = \frac{1}{2} [\theta_1 \quad \theta_2] \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} \theta_1 \\ \theta_2 \end{bmatrix} \quad (10.100)$$

where these two deflections θ_1 & θ_2 are the relative angular displacements between rods 1 and 2 & 2 and 3, respectively. From the geometry of the truss structure (and small angle approximations), one can form the following geometric constraints for these relative angular displacements as

$$\begin{aligned} \theta_1 &= \frac{Z_1 - Z_2}{x_1 - x_2} - \frac{Z_2 - Z_3}{x_2 + x_3} \\ &= \begin{bmatrix} \frac{1}{x_1 - x_2} & \left(\frac{-1}{x_1 - x_2} - \frac{1}{x_2 + x_3} \right) & \frac{1}{x_2 + x_3} & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} \\ &= C_1 \vec{Z} \end{aligned} \quad (10.101)$$

and

$$\begin{aligned}\theta_2 &= \frac{Z_3 - Z_2}{x_2 + x_3} - \frac{X_3 - X_4}{z_3 + z_4} \\ &= \begin{bmatrix} 0 & 0 & \frac{-1}{z_3+z_4} & \frac{1}{z_3+z_4} \end{bmatrix} \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} + \begin{bmatrix} 0 & \left(\frac{-1}{x_2+x_3} - \frac{1}{x_2+x_3}\right) & \frac{1}{x_2+x_3} & 0 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} \\ &= [C_2 \quad C_3] \begin{bmatrix} \vec{X} \\ \vec{Z} \end{bmatrix}\end{aligned}\quad (10.102)$$

Furthermore, assuming equal vibration displacements for the colinear masses, one has

$$\begin{aligned}X_1 &= X_2 = X_3 \\ \begin{bmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_3 \\ X_4 \end{bmatrix} \\ \vec{X} &= C'_x \vec{X}'\end{aligned}\quad (10.103)$$

$$\begin{aligned}Z_3 &= Z_4 \\ \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \\ Z_4 \end{bmatrix} &= \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ Z_3 \end{bmatrix} \\ \vec{Z} &= C'_z \vec{Z}'\end{aligned}\quad (10.104)$$

where the degrees of freedom of the truss has been reduced using these constraint matrices, resulting in the modified coordinate vectors, \vec{X}' and \vec{Z}' .

With these constraints, the kinetic energy of the truss can be rewritten as

$$\begin{aligned}T &= \frac{1}{2} \begin{bmatrix} \dot{\vec{X}}'^T & \dot{\vec{Z}}'^T \end{bmatrix} \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix}^T \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix} \begin{bmatrix} \dot{\vec{X}}' \\ \dot{\vec{Z}}' \end{bmatrix} \\ &= \frac{1}{2} \begin{bmatrix} \dot{\vec{X}}'^T & \dot{\vec{Z}}'^T \end{bmatrix} [MM]' \begin{bmatrix} \dot{\vec{X}}' \\ \dot{\vec{Z}}' \end{bmatrix}\end{aligned}\quad (10.105)$$

and the potential (or strain) energy of the truss can be rewritten as

$$\begin{aligned}
 U &= \frac{1}{2} [\vec{X}^T \quad \vec{Z}^T] \begin{bmatrix} 0 & C_1 \\ C_2 & C_3 \end{bmatrix}^T \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} 0 & C_1 \\ C_2 & C_3 \end{bmatrix} [\vec{X}] \\
 &= \frac{1}{2} [\vec{X}^T \quad \vec{Z}^T] K_c \begin{bmatrix} \vec{X} \\ \vec{Z} \end{bmatrix} \\
 &= \frac{1}{2} [\vec{X}'^T \quad \vec{Z}'^T \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix}^T] K_c \begin{bmatrix} C'_x & 0 \\ 0 & C'_z \end{bmatrix} \begin{bmatrix} \vec{X}' \\ \vec{Z}' \end{bmatrix} \\
 &= \frac{1}{2} [\vec{X}'^T \quad \vec{Z}'^T] K'_c \begin{bmatrix} \vec{X}' \\ \vec{Z}' \end{bmatrix}
 \end{aligned} \tag{10.106}$$

Finally, solving Lagrange's equation using these newly defined modified constrained mass matrix $[MM]'$ and modified constrained stiffness matrix K'_c , one has

$$[MM]' \begin{bmatrix} \ddot{\vec{X}} \\ \ddot{\vec{Z}} \end{bmatrix} + K'_c \begin{bmatrix} \vec{X} \\ \vec{Z} \end{bmatrix} = 0 \tag{10.107}$$

which has a modified constrained dynamic matrix $D'_c = [MM']^{-1}K'_c$. Thus, the same modal analysis for mutually orthogonal modes and coordinates previously performed can also be done for multi-directional motion as long as the “proper” mass and stiffness matrices are used.

10.2 Elastic Vehicle Mean-Axes

The perspective in this course on elastic-body dynamics is the ability to study the effects of elastic deformation on the flight dynamics of a vehicle, not to study purely structural/aeroelastic phenomena such as flutter or divergence. Thus, only the lower frequency modes are typically of interest for addition to the rigid-body modes developed in 6-DOF flight dynamics. Though all real vehicles are elastic, in many cases, this type of vibration analysis may only be necessary to check if the rigid-body assumption can be made for the modeling of the flight vehicle. In general, larger flight vehicles typically require some elastic modeling as the natural frequencies will be lower and more likely to interact with the rigid-body or flight controller modes of the vehicle.

The treatment for elastic vehicles will continue to make use of Lagrangian mechanics and energy concepts which will be stated here without proof, as typically these are developed in a graduate level dynamics course. These concepts will allow the derivation of important results for the reference frame requirements for elastic vehicles, namely the mean axes, which will be developed in this lecture. Previous sections have derived generalized coordinates to model vibration problems where the vibration modes are orthogonal to the rigid-body modes and demonstrated this for some simple lumped-mass models to assist in visualizing the construction of these elastic-body dynamic models. As such, these types of coordinate frames satisfy the mean-axis constraints which will be used for the mean-axes body-fixed frame derived in the equations of motion for describing elastic-body flight dynamics in the subsequent sections. Before developing the concept of mean-axes, first assume that the navigation frame N is inertial, i.e., the “flat-Earth” model.

Lagrangian Energies for Vehicles

Let the kinetic energy of the entire vehicle be

$$T = \frac{1}{2} \int_{Vol} \vec{v}_N \cdot \vec{v}_N \rho_V dV \quad (10.108)$$

where \vec{v}_N is the inertial velocity of a mass element of the vehicle, i.e.

$$\vec{v}_N = \vec{v}_{B/N,N} + \vec{v}_B + \vec{\omega}_{B/N} \times \vec{x}_B \quad (10.109)$$

where $\vec{v}_{B/N}$ is the velocity of the body-fixed frame relative to the navigation frame, \vec{v}_B is the velocity of the mass element in the body-fixed frame, $\vec{\omega}_{B/N}$ is the angular velocity of the body-fixed frame relative to the navigation frame, and \vec{x}_B is the position of the mass element in the body-fixed frame, i.e. relative to the center of mass. In addition, one has

$$\vec{x}_N = \vec{x}_B + \vec{x}_{B/N} \quad (10.110)$$

where $\vec{x}_{B/N}$ is the position of the origin of the body-fixed frame relative to the navigation frame. Then, by substitution, one has

$$\begin{aligned} T = \frac{1}{2} \int_{Vol} & [\vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \\ & + 2\vec{v}_{B/N,N} \cdot \vec{v}_B \\ & + 2(\vec{v}_{B/N,N} + \vec{v}_B) \cdot (\vec{\omega}_{B/N} \times \vec{x}_B) \\ & + \vec{v}_B \cdot \vec{v}_B \\ & + (\vec{\omega}_{B/N} \times \vec{x}_B) \cdot (\vec{\omega}_{B/N} \times \vec{x}_B)] \rho_V dV \end{aligned} \quad (10.111)$$

Furthermore, the potential energy of the vehicle includes gravitational potential energy U_g and elastic strain energy U_e . The gravitational potential energy for the vehicle can be modeled as

$$\begin{aligned} U_g &= - \int_{Vol} \vec{g} \cdot \vec{x}_N \rho_V dV \\ U_g &= - \int_{Vol} \vec{g} \cdot (\vec{x}_B + \vec{x}_{B/N}) \rho_V dV \end{aligned} \quad (10.112)$$

where \vec{g} is the acceleration due to gravity. The elastic strain energy is the energy stored in an elastic structure due to its deformation resulting from some applied force. The strain energy is the negative of the work done on the structure by the applied force, and the work is the force acting over a displacement \vec{d}_e . Thus, the position of a mass element of the vehicle can be represented as

$$\vec{x} = \vec{x}_{r-b} + \vec{d}_e \quad (10.113)$$

where \vec{x}_{r-b} is the position of the mass element in terms of its undeformed or rigid-body position. Furthermore, since $\vec{x}_{r-b,B}$ is invariant with respect to the body-fixed frame, one has

$$\vec{v}_B = \dot{\vec{d}}_{e,B} \quad (10.114)$$

Thus, using D'Alembert's principle to express the force on a mass element in terms of the mass of the element and its acceleration, one has for the elastic strain energy

$$U_e = -\frac{1}{2} \int_{Vol} \ddot{\vec{d}}_{e,B} \cdot \vec{d}_{e,B} \rho_V dV \quad (10.115)$$

Mean-Axes Body Frame

For rigid vehicle dynamics, the selection of body-fixed frame axes was arbitrary with the only requirement being the body-fixed frame origin as the vehicle's center of mass. Such a construction decoupled the rotation mode and the translation mode of the dynamics. For elastic vehicle dynamics, one must further consider the existence of any number of vibration modes which results in *additional* requirements for a body-fixed frame to exhibit decoupled dynamic modes, namely the **mean-axes constraints** which define coordinate axes about which the relative translational and angular momenta (about the center of mass) due to elastic vibrations are zero, i.e.

$$\int_{Vol} \dot{\vec{d}}_{e,B} \rho_V dV = \int_{Vol} \vec{x}_B \times \dot{\vec{d}}_{e,B} \rho_V dV = \vec{0} \quad (10.116)$$

This “special” **mean-axes body-fixed frame** can be shown to always exist for an elastic-body which will be assumed here and not proven. By substitution for these quantities from the relations in the previous section, one has

$$\int_{Vol} \frac{d}{dt} (\vec{x}_{r-b,B} + \vec{d}_{e,B}) \rho_V dV = \vec{0} \quad (10.117)$$

and

$$\int_{Vol} \vec{x}_{r-b,B} \times \dot{\vec{d}}_{e,B} \rho_V dV + \int_{Vol} \vec{d}_{e,B} \times \dot{\vec{d}}_{e,B} \rho_V dV = \vec{0} \quad (10.118)$$

which if the elastic displacement, $\vec{d}_{e,B}$, is sufficiently small such that only linear effects are considered, then one can neglect the moment from the elastic displacements, and one has the *practical mean-axes constraints* written as

$$\int_{Vol} \dot{\vec{d}}_{e,B} \rho_V dV = \int_{Vol} \vec{x}_{r-b} \times \dot{\vec{d}}_{e,B} \rho_V dV = \vec{0} \quad (10.119)$$

which are analogous to the modal orthogonality constraints for the lumped-mass systems considered earlier. Thus, while the mean-axes will be used for theoretical development, in practice, one uses the practical mean-axes constraints to confirm the selected axes are mean-axes through mutual orthogonality among all modes.

To demonstrate this, assume a free-vibration analysis has been performed as previously described yielding the n free-vibration mutually orthogonal mode shapes, \vec{v}_i , and frequencies, $\omega_i = \sqrt{\lambda_i}$, (i.e. eigenvectors and eigenvalues) including both the rigid-body and elastic modes. Then, the elastic vibrations can be expressed as

$$\vec{d}_{e,B} = \sum_{i=1}^n \vec{v}_i(\vec{x}) \eta_i(t) \quad (10.120)$$

where $\eta_i(t)$ is the generalized coordinate associated with the i -th vibration mode. In general, each mode shape, $\vec{v}_i(\vec{x})$, is a vector with components defined in the body-fixed frame with each component a function of the \vec{x} location on the *undeformed* structure. With this analysis, the *practical mean-axes constraints* can be rewritten as

$$\int_{Vol} \dot{\vec{d}}_{e,B} \rho_V dV = \sum_{i=1}^n \dot{\eta}_i(t) \left(\int_{Vol} \vec{v}_i(\vec{x}) \rho_V dV \right) = 0 \quad (10.121)$$

and

$$\int_{Vol} \vec{x}_{r-b} \times \dot{\vec{d}}_{e,B} \rho_V dV = \sum_{i=1}^n \left(\int_{Vol} \vec{x}_{r-b} \times \vec{v}_i(\vec{x}) \rho_V dV \right) = 0 \quad (10.122)$$

which are satisfied as the integrals inside the parentheses above correspond to the momenta conservation requirements and the selected vibration modes are, by design, orthogonal to the rigid-body translation and rotation modes (with respect to the mass distribution here).

Mean-Axis Constraints on Energies

Now, one can apply the *mean-axis constraints* to the energies to greatly simplify these equations. For the first term of the kinetic energy, as the velocity of the center of mass is independent of the volume, one has

$$\begin{aligned} \int_{Vol} \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \rho_V dV &= \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \int_{Vol} \rho_V dV \\ &= m \vec{v}_{B/N,N} \cdot \vec{v}_{B/N,N} \end{aligned} \quad (10.123)$$

where m is the total mass of the vehicle. The second term of the kinetic energy becomes zero as

$$\int_{Vol} \vec{v}_{B/N,N} \cdot \vec{d}_{e,B} \rho_V dV = \vec{v}_{B/N,N} \cdot \int_{Vol} \vec{d}_{e,B} \rho_V dV = 0 \quad (10.124)$$

For the third term of the kinetic energy, as the requirement for the origin of the mean-axis body-fixed frame be the instantaneous center of mass states that

$$\int_{Vol} \vec{x}_B \rho_V dV = 0 \quad (10.125)$$

and note that

$$\int_{Vol} \vec{d}_{e,B} \cdot (\vec{\omega}_{B/N} \times \vec{x}_B) \rho_V dV = \left(\int_{Vol} \vec{x}_B \times \vec{v}_{B/N} \rho_V dV \right) \cdot \vec{\omega}_{B/N} = 0 \quad (10.126)$$

Thus, one has

$$\int_{Vol} \left(\vec{v}_{B/N,N} + \vec{d}_{e,B} \right) \cdot (\vec{\omega}_{B/N} \times \vec{x}_B) \rho_V dV = \vec{v}_{B/N,N} \cdot \left(\vec{\omega}_{B/N} \times \int_{Vol} \vec{x}_B \rho_V dV \right) = 0 \quad (10.127)$$

The fourth term of the kinetic energy can be rewritten using the mode shapes and generalized coordinate summations for the displacement *velocity*, i.e.

$$\begin{aligned} \int_{Vol} \vec{v}_B \cdot \vec{v}_B \rho_V dV &= \frac{1}{2} \int_{Vol} \left(\sum_{i=1}^n \vec{v}_i(\vec{x}) \dot{\eta}_i \cdot \sum_{i=1}^n \vec{v}_i(\vec{x}) \dot{\eta}_i \right) \rho_V dV \\ &= \int_{Vol} \left(\sum_{i=1}^n \vec{v}_i(\vec{x}) \cdot \vec{v}_i(\vec{x}) \dot{\eta}_i^2 \right) \rho_V dV \\ &= \sum_{i=1}^n \mathcal{M}_i \dot{\eta}_i^2 \end{aligned} \quad (10.128)$$

where \mathcal{M}_i is the i -th generalized mass of the i -th vibration mode. This simplification is due to the mutual orthogonality of the vibration modes, i.e.

$$\int_{Vol} \vec{v}_i \cdot \vec{v}_j \rho_V dV = \begin{cases} 0 & i \neq j \\ \mathcal{M}_i & i = j \end{cases} \quad (10.129)$$

The fifth term of the kinetic energy can be rewritten in terms of the inertia matrix as

$$\int_{Vol} (\vec{\omega}_{B/N} \times \vec{x}_B) \cdot (\vec{\omega}_{B/N} \times \vec{x}_B) \rho_V dV = \vec{\omega}_{B/N}^T I_G \vec{\omega}_{B/N} \quad (10.130)$$

Thus, the kinetic energy can be rewritten as

$$T = \frac{1}{2} m \vec{v}_{B/N,N}^T \vec{v}_{B/N,N} + \frac{1}{2} \vec{\omega}_{B/N}^T I_G \vec{\omega}_{B/N} + \frac{1}{2} \sum_{i=1}^n \mathcal{M}_i \dot{\eta}_i^2 \quad (10.131)$$

Next, applying the *mean-axis constraints* to the gravitational potential energy, one has

$$\begin{aligned} U_g &= - \int_{Vol} \vec{g} \cdot (\vec{x}_B + \vec{x}_{B/N}) \rho_V dV \\ U_g &= - \vec{g} \cdot \int_{Vol} \vec{x}_B \rho_V dV - \vec{g} \cdot \vec{x}_{B/N} \int_{Vol} \rho_V dV \\ U_g &= -m \vec{g}^T \vec{x}_{B/N} \end{aligned} \quad (10.132)$$

Finally, applying the *mean-axis constraints* to the elastic strain energy, one has

$$U_e = -\frac{1}{2} \int_{Vol} \sum_{i=1}^n \vec{v}_i(\vec{x}) \ddot{\eta}_i(t) \cdot \sum_{i=1}^n \vec{v}_i(\vec{x}) \eta_i(t) \rho_V dV \quad (10.133)$$

and due to the orthogonality of the modes, one has

$$U_e = -\frac{1}{2} \int_{Vol} \sum_{i=1}^n \vec{v}_i(\vec{x}) \cdot \vec{v}_i(\vec{x}) \ddot{\eta}_i(t) \eta_i(t) \rho_V dV \quad (10.134)$$

Then, recalling the sinusoidal solution for the modal coordinates, i.e.

$$\eta_i(t) = A_i \cos(\omega_i t + \Gamma_i) \quad (10.135)$$

one has

$$U_e = -\frac{1}{2} \sum_{i=1}^n \omega_i^2 \eta_i^2(t) \mathcal{M}_i \quad (10.136)$$

Generalized Coordinate Selection

Before applying Lagrange's equation to the energies to obtain the elastic vehicle equations of motion, one must select suitable generalized coordinates. First, one can select the inertial position of the body-fixed frame's origin (i.e. the center of mass) as the coordinates

$$\vec{x}_{B/N} = \begin{bmatrix} x_{B/N} \\ y_{B/N} \\ z_{B/N} \end{bmatrix} \quad (10.137)$$

which was represented in rigid vehicle dynamics without the referencing slash which is appropriate for rigid bodies as the relative position of any point on the body can be represented with the location of the center of

mass and the attitude of the vehicle. Though for elastic vehicles this is no longer the case, the B/N subscript will be dropped to easily compare the elastic vehicle dynamics with the rigid-body dynamics, i.e.

$$\vec{x}_{B/N,N} = \begin{bmatrix} x, N \\ y, N \\ z, N \end{bmatrix} \quad (10.138)$$

will be the selected generalized coordinates. Also recall that $z_N = -h$ for the flat-Earth approximation. Thus, the velocity of the body-fixed frame's origin expressed in the navigation frame is

$$\vec{v}_{B/N,N} = \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} \quad (10.139)$$

and the body-fixed frame axes as

$$\vec{v}_{B/N,B} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (10.140)$$

Secondly, one can select the angular velocity of the body-fixed frame relative to the navigation frame as

$$\vec{\omega}_{B/N} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (10.141)$$

which can also be expressed in either the body-fixed frame or the navigation frame (with an appropriate subscript). The generalized coordinates for this derivation will use the body-fixed frame axes. Thirdly, one can select the 3-2-1 Euler angles ψ, θ, ϕ to define the orientation of the mean-axes body-fixed frame with respect to the navigation frame. Recall that the Euler angles are related to the angular velocity by the equations

$$\begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\sin \theta \\ 0 & \cos \phi & -\sin \phi \cos \theta \\ 0 & -\sin \phi & \cos \phi \cos \theta \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (10.142)$$

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (10.143)$$

Thus, one can select the generalized coordinates as

$$\vec{q} = [x \ y \ z \ \phi \ \theta \ \psi \ \eta_i, i = 1, 2, \dots]^T \quad (10.144)$$

which is the same as for rigid vehicle dynamics with the addition of the modal coordinates corresponding to the mutually orthogonal mode shapes.

Using these definitions, one can write the kinetic energy of the elastic vehicle as

$$T = \frac{1}{2}m [\dot{x}_N \ \dot{y}_N \ \dot{z}_N] \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} + \frac{1}{2} [p \ q \ r] I_G \begin{bmatrix} p \\ q \\ r \end{bmatrix} + \frac{1}{2} \sum_{i=1}^n M_i \dot{\eta}_i^2 \quad (10.145)$$

the gravitational potential energy as

$$U_g = -mgz_N = mgh \quad (10.146)$$

and the elastic strain energy remains the same.

$$U_e = -\frac{1}{2} \sum_{i=1}^n \omega_i^2 \eta_i^2(t) \mathcal{M}_i \quad (10.147)$$

Using Lagrange's equation in vector form, i.e.,

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\vec{q}}} \right) - \frac{\partial T}{\partial \vec{q}} + \frac{\partial U}{\partial \vec{q}} = \vec{Q}^T = \frac{\partial \delta W}{\partial \delta q} \quad (10.148)$$

one can derive the equations of motion for elastic flight vehicles in three coupled vector equations of motion: translation, rotation, and vibration.

10.3 Elastic Aerospace Vehicle Dynamics

Translation Equations of Motion

For a constant-mass vehicle, consider the inertial center of mass coordinates, $\vec{x}_N = [x_N \ y_N \ z_N]^T$, chosen as the translation generalized coordinates, and applying Lagrange's equation for the translational kinetic energy, i.e.

$$T_{tran} = \frac{1}{2}m \begin{bmatrix} \dot{x}_N & \dot{y}_N & \dot{z}_N \end{bmatrix} \begin{bmatrix} \ddot{x}_N \\ \ddot{y}_N \\ \ddot{z}_N \end{bmatrix} \quad (10.149)$$

one has

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\vec{x}}_N} \right) - \frac{\partial T}{\partial \vec{x}_N} = m \begin{bmatrix} \ddot{x}_N \\ \ddot{y}_N \\ \ddot{z}_N \end{bmatrix} = m \ddot{\vec{x}}_N \quad (10.150)$$

and for the gravitational potential energy, one has

$$\frac{\partial U_g}{\partial \vec{x}_N} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} \quad (10.151)$$

Including the generalized forces in the navigation frame \vec{Q}_N , one has the following translation EOMs

$$\ddot{\vec{x}}_N = \begin{bmatrix} \ddot{x}_N \\ \ddot{y}_N \\ \ddot{z}_N \end{bmatrix} = \begin{bmatrix} \frac{Q_{x,N}}{m} \\ \frac{Q_{y,N}}{m} \\ \frac{Q_{z,N}}{m} + g \end{bmatrix} \quad (10.152)$$

However, for flight dynamics, one typically desires to convert this to the body frame accelerations and velocities, i.e. recalling the conversion of the velocity by

$$\ddot{\vec{x}}_N = \ddot{\vec{x}}_B + \vec{\omega}_{B/N} \times \dot{\vec{x}}_B \quad (10.153)$$

one has using the definitions for the body frame linear and angular velocity components

$$\begin{aligned}\ddot{\vec{x}}_N &= \begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} + \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} u \\ v \\ w \end{bmatrix} \\ &= \begin{bmatrix} \dot{u} - rv + qw \\ \dot{v} + ru - wp \\ \dot{w} - qu + pv \end{bmatrix}\end{aligned}\quad (10.154)$$

For the generalized forces for flight vehicle, recall that the net force (besides gravitational) are the net aerodynamic and propulsive which are defined in the body frame as

$$\vec{F}_{a,B} + \vec{F}_{p,B} = \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} \quad (10.155)$$

The virtual work, δW_F , is done by these forces as virtual displacements in the body frame, i.e. $\delta \vec{x}_B = [\delta x_B \ \delta y_B \ \delta z_B]^T$, which can be written as

$$\delta W_F = [mX \ mY \ mZ] \begin{bmatrix} \delta x_B \\ \delta y_B \\ \delta z_B \end{bmatrix} \quad (10.156)$$

In terms of the selected generalized coordinates, one can use the DCM from $N \rightarrow B$ which is a function of the Euler angles as follows:

$$\delta W_F = [mX \ mY \ mZ] C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} \delta x_N \\ \delta y_N \\ \delta z_N \end{bmatrix} \quad (10.157)$$

Thus the generalized forces become

$$\begin{aligned}\begin{bmatrix} Q_{x,N} \\ Q_{y,N} \\ Q_{z,N} \end{bmatrix} &= \frac{\partial \delta W_F}{\partial \delta \vec{x}_N} = [mX \ mY \ mZ] C_{B \leftarrow N}(\phi, \theta, \psi) \\ &= C_{B \leftarrow N}^T(\phi, \theta, \psi) \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix}\end{aligned}\quad (10.158)$$

which can be rewritten in body frame coordinates as simply

$$\begin{bmatrix} Q_{x,B} \\ Q_{y,B} \\ Q_{z,B} \end{bmatrix} = C_{B \leftarrow N}(\phi, \theta, \psi) C_{B \leftarrow N}^T(\phi, \theta, \psi) \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} = \begin{bmatrix} mX \\ mY \\ mZ \end{bmatrix} \quad (10.159)$$

as expected. Lastly, for the gravitational force, one has in the body frame coordinates

$$\begin{bmatrix} -mg \sin \theta \\ mg \cos \theta \sin \phi \\ mg \cos \theta \cos \phi \end{bmatrix} = C_{B \leftarrow N}(\phi, \theta, \psi) \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix} \quad (10.160)$$

Thus, one has the same rigid vehicle translation equations of motion for elastic vehicle translation equations of motion, i.e.

$$\begin{bmatrix} \dot{u} - rv + qw \\ \dot{v} + ru - wp \\ \dot{w} - qu + pv \end{bmatrix} = \begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \end{bmatrix} \quad (10.161)$$

and any elastic deformation effects will enter by the aerodynamic and propulsive forces.

Rotation Equations of Motion

For a constant-mass vehicle, consider the Euler angles, $\vec{q}_\perp = [\phi \ \theta \ \psi]^T$, chosen as the rotation generalized coordinates. However as the Euler angles represent three sequential rotations, the relationship between the body frame angular velocity (which appears in the kinetic energy) and the Euler angles is

$$\vec{\omega}_{B/N} = \begin{bmatrix} p \\ q \\ r \end{bmatrix} = C_\omega(\vec{q}_\perp) \dot{\vec{q}}_\perp = \begin{bmatrix} 1 & 0 & -\sin \phi \\ 0 & \cos \phi & \cos \theta \sin \phi \\ 0 & -\sin \phi & \cos \theta \cos \phi \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \quad (10.162)$$

Thus, one has

$$\frac{\partial \vec{\omega}_{B/N}}{\partial \dot{\vec{q}}_\perp} = C_\omega \quad (10.163)$$

Next, note that the rotational kinetic energy becomes

$$T_{rot} = \frac{1}{2} \vec{\omega}_{B/N}^T I_G \vec{\omega}_{B/N} \quad (10.164)$$

Thus, one has

$$\frac{\partial T}{\partial \vec{\omega}_{B/N}} = \vec{\omega}_{B/N}^T I_G \quad (10.165)$$

Furthermore, one can show

$$\frac{\partial \vec{\omega}_{B/N}}{\partial \vec{q}_\perp} = \begin{bmatrix} 0 & -\dot{\psi} \cos \theta & 0 \\ \dot{\psi} \cos \theta \cos \phi - \dot{\theta} \sin \phi & -\dot{\psi} \sin \theta \sin \phi & 0 \\ -\dot{\psi} \cos \theta \sin \phi - \dot{\theta} \cos \phi & -\dot{\psi} \sin \theta \cos \phi & 0 \end{bmatrix} = \begin{bmatrix} 0 & -\dot{\psi} \cos \theta & 0 \\ r & -\dot{\psi} \sin \theta \sin \phi & 0 \\ -q & -\dot{\psi} \sin \theta \cos \phi & 0 \end{bmatrix} \quad (10.166)$$

Then, applying Lagrange's equation for the kinetic energy, one has

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\vec{q}}_\perp} - \frac{\partial T}{\partial \vec{q}_\perp} = \frac{d}{dt} \left(\frac{\partial T}{\partial \vec{\omega}_{B/N}} \frac{\partial \vec{\omega}_{B/N}}{\partial \dot{\vec{q}}_\perp} \right) - \frac{\partial T}{\partial \vec{\omega}_{B/N}} \frac{\partial \vec{\omega}_{B/N}}{\partial \vec{q}_\perp} \quad (10.167)$$

which after the matrix multiplications and some algebra, it can be shown

$$\frac{d}{dt} \frac{\partial T}{\partial \dot{\vec{q}}_\perp} - \frac{\partial T}{\partial \vec{q}_\perp} = C_\omega^T (I_G \dot{\vec{\omega}}_{B/N} + \vec{\omega}_{B/N} \times I_G \vec{\omega}_{B/N}) = \vec{Q}_\perp \quad (10.168)$$

where the virtual work associated with the net moment on the flight vehicle in the body frame can be related to the virtual angular displacements simply by

$$\delta W_M = [I_{xx}L \quad I_{yy}M \quad I_{zz}N] \begin{bmatrix} \delta \phi \\ \delta \theta \\ \delta \psi \end{bmatrix} \quad (10.169)$$

where the infinitesimal rotations due to the angular displacements are related to the Euler angles themselves, by

$$\begin{bmatrix} \delta\phi \\ \delta\theta \\ \delta\psi \end{bmatrix} = C_\omega \vec{q}_\perp \quad (10.170)$$

Thus, in terms of the generalized coordinates

$$\vec{Q}_\perp^T = \frac{\partial \delta W_M}{\partial \vec{q}_\perp} = [I_{xx}L \quad I_{yy}M \quad I_{zz}N] C_\omega \quad (10.171)$$

$$\vec{Q}_\perp = C_\omega^T \begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} \quad (10.172)$$

Thus, one has the same rigid vehicle rotation equations of motion for elastic vehicle rotation equations of motion, i.e.

$$I_G \dot{\omega}_{B/N} + \vec{\omega}_{B/N} \times I_G \vec{\omega}_{B/N} = \begin{bmatrix} I_{xx}L \\ I_{yy}M \\ I_{zz}N \end{bmatrix} \quad (10.173)$$

which for $I_{xy} = I_{yz} = 0$ can be written out as

$$\begin{bmatrix} \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} = \begin{bmatrix} L \\ M \\ N \end{bmatrix} \quad (10.174)$$

and any elastic deformation effects will enter by the aerodynamic and propulsive moments.

Vibration Equations of Motion

For a constant-mass vehicle, consider the n vibration coordinates, $\vec{\eta} = [\eta_1 \cdots \eta_n]^T$, chosen as the vibration generalized coordinates. Applying Lagrange's equation for the vibration kinetic energy and elastic strain energy for each individual modal coordinate, one has

$$\frac{d}{dt} \left(\frac{\partial T}{\partial \dot{\eta}_i} \right) - \frac{\partial T}{\partial \eta_i} + \frac{\partial U_e}{\partial \eta_i} = Q_i = \frac{\partial \delta W}{\partial \delta \eta_i} \quad (10.175)$$

one has n equations of motion for the vibration coordinates of the form

$$\ddot{\eta}_i + \omega_i^2 \eta_i = \frac{Q_i}{M_i} \quad i = 1, \dots, n \quad (10.176)$$

where M is the generalized mass and Q_i is the generalized force, each associated with the i -th vibration mode.

For aerodynamics modeling for the generalized mass, let the external pressure distribution acting on the surface of the vehicle's structure at point \vec{x}_B be defined as $\vec{P}(\vec{x}_B)$. By construction, the integral of this

pressure distribution will result in the aerodynamic and propulsive forces and moments X, Y, Z, L, M, N . Then, the local elastic deformation of the structure can be written in terms of the modes as

$$\delta d_e(\vec{x}_B) = \sum_{i=1}^n \vec{v}_i \delta \eta_i(t) \quad (10.177)$$

Thus, the virtual work done due to the pressure \vec{P} located at \vec{x}_B on the structure is

$$d\delta W_P = \vec{P}(\vec{x}_B) \cdot \sum_{i=1}^n \vec{v}_i(\vec{x}_B) \delta \eta_i(t) dS \quad (10.178)$$

where dS is the infinitesimal surface area over which the pressure applies. Thus, the total virtual work is

$$\begin{aligned} \delta W_P &= \int_{Area} \vec{P}(\vec{x}_B) \cdot \sum_{i=1}^n \vec{v}_i(\vec{x}_B) \delta \eta_i(t) dS \\ &= \sum_{i=1}^n \int_{Area} \vec{P}(\vec{x}_B) \cdot \vec{v}_i(\vec{x}_B) dS \delta \eta_i(t) \end{aligned} \quad (10.179)$$

Finally, the n vibration equations of motion become

$$\ddot{\eta}_i + \omega_i^2 \eta_i = \frac{1}{M_i} \int_{Area} \vec{P}(\vec{x}_B) \cdot \vec{v}_i(\vec{x}_B) dS \quad i = 1, \dots, n \quad (10.180)$$

which connects the aerodynamic pressure with the structural deformations.

Point Motion on Elastic Vehicle

With these three sets of equations of motion, one is now able to describe the motion of any point on the elastic flight vehicle which is a combination of the rigid body motion *and* the vibration motion. Specifically, the position of any point on the elastic vehicle as defined in the navigation frame is a linear combination of the body frame's origin (i.e. the center of mass), the rigid body position of the point, \vec{x}_{r-b} , and the elastic displacement, d_e , i.e.

$$\vec{x}_N = \vec{x}_{B/N} + \vec{x}_{r-b} + d_e(\vec{x}_B, t) \quad (10.181)$$

or

$$\vec{x}_N = \vec{x}_{B/N} + \vec{x}_{r-b} + \sum_{i=1}^n \vec{v}_i(\vec{x}_B) \eta_i(t) \quad (10.182)$$

where each of these terms can be determined by the elastic vehicle equations of motion (assuming one has a solution for the free-vibration problem for the mode shapes). Note that here the position of the instantaneous center of mass is governed by the rigid body translation equations of motion which typically uses the body frame coordinates for the velocity of the body frame, thus, one should recall

$$\dot{\vec{x}}_{B/N} = \begin{bmatrix} \dot{x}_N \\ \dot{y}_N \\ \dot{z}_N \end{bmatrix} = C_{N \leftarrow B} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (10.183)$$

based on the final equations given previously.

Elastic Flight Vehicle Dynamics

Thus, the elastic flight vehicle equations of motion can be written together as

$$\begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \\ \dot{p} + \frac{I_{zz}-I_{yy}}{I_{xx}}qr - \frac{I_{xz}}{I_{xx}}(\dot{r} + pq) \\ \dot{q} + \frac{I_{xx}-I_{zz}}{I_{yy}}pr - \frac{I_{xz}}{I_{yy}}(r^2 - p^2) \\ \dot{r} + \frac{I_{yy}-I_{xx}}{I_{zz}}pq - \frac{I_{xz}}{I_{zz}}(\dot{p} - qr) \end{bmatrix} = \begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} \quad (10.184)$$

$$\ddot{\eta}_i + 2\zeta_i \omega_i \dot{\eta}_i + \omega_i^2 \eta_i = \frac{Q_i}{M_i}, \quad i = 1, \dots, n$$

And defining the linear and angular velocity of the body frame as the rigid state vector

$$\vec{x}_{rig} = [u \quad v \quad w \quad p \quad q \quad r]^T \quad (10.185)$$

the modal coordinates and modal coordinate rates as the vibration state vector

$$\vec{x}_{vib} = [\eta_1 \quad \dots \quad \eta_n \quad \dot{\eta}_1 \quad \dots \quad \dot{\eta}_n]^T \quad (10.186)$$

With these definitions, the nonlinear equations of motion for an elastic flight vehicle can be rewritten as

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{I_{xz}}{I_{zz}} \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{I_{xz}}{I_{zz}} & 0 & 1 \end{bmatrix} \vec{x}_{rig} = \begin{bmatrix} rv - qw - g \sin \theta \\ pw - ru + g \cos \theta \sin \phi \\ qu - pv + g \cos \theta \cos \phi \\ \frac{I_{yy}-I_{zz}}{I_{xx}}qr + \frac{I_{xz}}{I_{xx}}pq \\ \frac{I_{zz}-I_{xx}}{I_{yy}}pr + \frac{I_{xz}}{I_{yy}}(r^2 - p^2) \\ \frac{I_{xx}-I_{yy}}{I_{zz}}pq - \frac{I_{xz}}{I_{zz}}qr \end{bmatrix} + \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} \quad (10.187)$$

$$\vec{x}_{vib} = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ -\Omega^2 & -2\Omega_\zeta \end{bmatrix} \vec{x}_{vib} + \begin{bmatrix} \vec{0}_n \\ \frac{Q_1}{M_1} \\ \vdots \\ \frac{Q_n}{M_n} \end{bmatrix}$$

where

$$\Omega^2 = \begin{bmatrix} \omega_1^2 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \omega_n^2 \end{bmatrix} \quad (10.188)$$

and

$$\Omega_\zeta = \begin{bmatrix} \zeta_1 \omega_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \zeta_n \omega_n \end{bmatrix} \quad (10.189)$$

where ζ has been added as potential damping terms to the equations of motion here.

Then, recalling that the aerodynamic forces and moments and the generalized forces can both be modeled as linear functions of the rigid states, vibration states, and control inputs, one can define the following terms:

$$\mathcal{I} = \begin{bmatrix} 1 & 0 & -X_{\dot{w}} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 - Z_{\dot{w}} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -\frac{I_{xz}}{I_{zz}} \\ 0 & 0 & -M_{\dot{w}} & 0 & 1 & 0 \\ 0 & 0 & 0 & -\frac{I_{xz}}{I_{zz}} & 0 & 1 \end{bmatrix} \quad (10.190)$$

(note if $I_{xz} = 0$ and one ignores \dot{w} effects, \mathcal{I} will be a 6×6 identity matrix),

$$\mathcal{M} = \begin{bmatrix} \mathcal{M}_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \mathcal{M}_n \end{bmatrix} \quad (10.191)$$

$$f_{rig}(\vec{x}_{rig}, \theta, \phi) = \mathcal{I}^{-1} \begin{bmatrix} rv - qw - g \sin \theta + X_0 \\ pw - ru + g \cos \theta \sin \phi + Y_0 \\ qu - pv + g \cos \theta \cos \phi + Z_0 \\ \frac{I_{xx} - I_{zz}}{I_{xx}} qr + \frac{I_{xz}}{I_{xx}} pq + L_0 \\ \frac{I_{zz} - I_{xx}}{I_{yy}} pr + \frac{I_{xz}}{I_{yy}} (r^2 - p^2) + M_0 \\ \frac{I_{xx} - I_{yy}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} qr + N_0 \end{bmatrix} \quad (10.192)$$

$$\mathcal{A}_{rig \leftarrow rig} = \mathcal{I}^{-1} \begin{bmatrix} X_u & 0 & X_w & 0 & X_q & 0 \\ 0 & Y_v & 0 & Y_p & 0 & Y_r \\ Z_u & 0 & Z_w & 0 & Z_q & 0 \\ 0 & L_v & 0 & L_p & 0 & L_r \\ M_u & 0 & M_w & 0 & M_q & 0 \\ 0 & N_v & 0 & N_p & 0 & N_r \end{bmatrix} \quad (10.193)$$

$$\mathcal{A}_{rig \leftarrow \eta} = \mathcal{I}^{-1} \begin{bmatrix} X_{\eta_1} & \cdots & X_{\eta_n} \\ Y_{\eta_1} & \cdots & Y_{\eta_n} \\ Z_{\eta_1} & \cdots & Z_{\eta_n} \\ L_{\eta_1} & \cdots & L_{\eta_n} \\ M_{\eta_1} & \cdots & M_{\eta_n} \\ N_{\eta_1} & \cdots & N_{\eta_n} \end{bmatrix} \quad (10.194)$$

$$\mathcal{A}_{rig \leftarrow \dot{\eta}} = \mathcal{I}^{-1} \begin{bmatrix} X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_n} \\ Y_{\dot{\eta}_1} & \cdots & Y_{\dot{\eta}_n} \\ Z_{\dot{\eta}_1} & \cdots & Z_{\dot{\eta}_n} \\ L_{\dot{\eta}_1} & \cdots & L_{\dot{\eta}_n} \\ M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_n} \\ N_{\dot{\eta}_1} & \cdots & N_{\dot{\eta}_n} \end{bmatrix} \quad (10.195)$$

$$\mathcal{B}_{rig} = \mathcal{I}^{-1} \begin{bmatrix} 0 & X_{\delta_e} & 0 & X_{\delta_t} \\ 0 & 0 & Y_{\delta_r} & 0 \\ 0 & Z_{\delta_e} & 0 & Z_{\delta_t} \\ L_{\delta_a} & 0 & L_{\delta_r} & 0 \\ 0 & M_{\delta_e} & 0 & M_{\delta_t} \\ N_{\delta_a} & 0 & N_{\delta_r} & 0 \end{bmatrix} \quad (10.196)$$

$$A_{vib \leftarrow rig} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_u} & Q_{1_v} & Q_{1_w} & Q_{1_p} & Q_{1_q} & Q_{1_r} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ Q_{n_u} & Q_{n_v} & Q_{n_w} & Q_{n_p} & Q_{n_q} & Q_{n_r} \end{bmatrix} \quad (10.197)$$

$$A_{vib \leftarrow \eta} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_{\eta_1}} & \cdots & Q_{1_{\eta_n}} \\ \vdots & \ddots & \vdots \\ Q_{n_{\eta_1}} & \cdots & Q_{n_{\eta_n}} \end{bmatrix} - \Omega^2 \quad (10.198)$$

$$A_{vib \leftarrow \dot{\eta}} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_{\dot{\eta}_1}} & \cdots & Q_{1_{\dot{\eta}_n}} \\ \vdots & \ddots & \vdots \\ Q_{n_{\dot{\eta}_1}} & \cdots & Q_{n_{\dot{\eta}_n}} \end{bmatrix} - 2\Omega_\zeta \quad (10.199)$$

and

$$B_{vib} = \mathcal{M}^{-1} \begin{bmatrix} Q_{1_{\delta_a}} & Q_{1_{\delta_e}} & Q_{1_{\delta_r}} & Q_{1_{\delta_t}} \\ \vdots & \vdots & \vdots & \vdots \\ Q_{n_{\delta_a}} & Q_{n_{\delta_e}} & Q_{n_{\delta_r}} & Q_{n_{\delta_t}} \end{bmatrix} \quad (10.200)$$

With these definitions, one may finally rewrite the elastic flight vehicle equations of motion in state-space form as

$$\begin{aligned} \dot{\vec{x}}_{rig} &= f_{rig}(\vec{x}_{rig}, \phi, \theta) + \mathcal{A}_{rig \leftarrow rig} \vec{x}_{rig} + [\mathcal{A}_{rig \leftarrow \eta} \quad \mathcal{A}_{rig \leftarrow \dot{\eta}}] \vec{x}_{vib} + \mathcal{B}_{rig} \vec{u} \\ \dot{\vec{x}}_{vib} &= \begin{bmatrix} 0_{n \times 6} \\ A_{vib \leftarrow rig} \end{bmatrix} \vec{x}_{rig} + \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ A_{vib \leftarrow \eta} & A_{vib \leftarrow \dot{\eta}} \end{bmatrix} \vec{x}_{vib} + \begin{bmatrix} 0_{n \times 4} \\ B_{vib} \end{bmatrix} \vec{u} \end{aligned} \quad (10.201)$$

and it should be noted that here, v , w , and \dot{w} have been used in place of β , α , and $\dot{\alpha}$, respectively, though these could be replaced by the linear approximations and coefficient conversions if required. Note also that one would also require the supplemental Euler angle equation to relate p , q , and r to $\dot{\phi}$ and $\dot{\theta}$ to complete the state-space formulation, i.e.

$$\begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} 1 & \sin \phi \tan \theta & \cos \phi \tan \theta \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi \sec \theta & \cos \phi \sec \theta \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \quad (10.202)$$

10.4 Elastic Airplane Dynamics

Dynamic-Elastic Airplane Effects

Previous sections have provided the background for developing the elastic vibration equations of motion (EOMs) alongside the rigid-body EOMs. These showed that the vibration modes have their own set of

equations of motion and enter the rigid body EOMs only through elastic effects on the aerodynamic and propulsive forces and moments. These considerations can be studied as **dynamic-elastic effects**, which models the elastic effects on these forces and moments in the rigid body EOMs and the vibration dynamics, or the **static-elastic effects** which *only* consider the elastic deformation effects on the forces and moments in the rigid body EOMs and not do include the vibration dynamics. This section will consider the dynamic-elastic effects modeling for both the nonlinear forms of airplane dynamics while the subsequent section will discuss the linearized dynamic-elastic effects and the static-elastic effects modeling for airplane dynamics.

To this end, recall the rigid airplane equations of motion

$$\begin{bmatrix} X - g \sin \theta \\ Y + g \cos \theta \sin \phi \\ Z + g \cos \theta \cos \phi \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} \dot{u} + qw - rv \\ \dot{v} + ru - pw \\ \dot{w} + pv - qu \\ \dot{p} + \frac{I_{zz} - I_{yy}}{I_{xx}} qr - \frac{I_{xz}}{I_{xx}} (\dot{r} + pq) \\ \dot{q} + \frac{I_{xx} - I_{zz}}{I_{yy}} pr - \frac{I_{xz}}{I_{yy}} (r^2 - p^2) \\ \dot{r} + \frac{I_{yy} - I_{xx}}{I_{zz}} pq - \frac{I_{xz}}{I_{zz}} (\dot{p} - qr) \end{bmatrix} \quad (10.203)$$

where the body frame forces can alternatively be written using the thrust, \vec{T} , and the wind frame aerodynamic forces: lift L , side S , and drag D , as

$$\begin{aligned} m \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= \vec{T} + C_{B \leftarrow W}(\alpha, \beta) \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \\ m \begin{bmatrix} X \\ Y \\ Z \end{bmatrix} &= \vec{T} + \begin{bmatrix} \cos \alpha \cos \beta & -\cos \alpha \sin \beta & -\sin \alpha \\ \sin \beta & \cos \beta & 0 \\ \sin \alpha \cos \beta & -\sin \alpha \sin \beta & \cos \alpha \end{bmatrix} \begin{bmatrix} -D \\ S \\ -L \end{bmatrix} \end{aligned} \quad (10.204)$$

where the dynamic-elastic effects can be alternatively applied to L , S , and D instead of X , Y , Z . In addition, recall that one can model these aerodynamic and propulsive forces using stability and control derivatives/coefficients (where $\delta_T = T$). The same approach can be used to add coefficients for each modal coordinate and modal coordinate rate, i.e.

$$\begin{aligned} \begin{bmatrix} X \\ Z \\ M \end{bmatrix} &= \begin{bmatrix} X_0 \\ Z_0 \\ M_0 \end{bmatrix} + \begin{bmatrix} 0 & X_{\dot{\alpha}} & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \dot{u} \\ \dot{\alpha} \\ \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_{\alpha} & X_q \\ Z_u & Z_{\alpha} & Z_q \\ M_u & M_{\alpha} & M_q \end{bmatrix} \begin{bmatrix} u \\ \alpha \\ q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \delta_e \\ \delta_t \end{bmatrix} \\ &\quad + \begin{bmatrix} X_{\dot{\eta}_1} & \dots & X_{\dot{\eta}_1} \\ Z_{\dot{\eta}_1} & \dots & Z_{\dot{\eta}_1} \\ M_{\dot{\eta}_1} & \dots & M_{\dot{\eta}_1} \end{bmatrix} \begin{bmatrix} \dot{\eta}_1 \\ \vdots \\ \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} X_{\eta_1} & \dots & X_{\eta_1} \\ Z_{\eta_1} & \dots & Z_{\eta_1} \\ M_{\eta_1} & \dots & M_{\eta_1} \end{bmatrix} \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix} \end{aligned} \quad (10.205)$$

and

$$\begin{aligned} \begin{bmatrix} Y \\ L \\ N \end{bmatrix} &= \begin{bmatrix} Y_0 \\ L_0 \\ N_0 \end{bmatrix} + \begin{bmatrix} Y_{\beta} & Y_p & Y_r \\ L_{\beta} & L_p & L_r \\ N_{\beta} & N_p & N_r \end{bmatrix} \begin{bmatrix} \beta \\ p \\ r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \delta_a \\ \delta_r \end{bmatrix} \\ &\quad + \begin{bmatrix} Y_{\dot{\eta}_1} & \dots & Y_{\dot{\eta}_1} \\ L_{\dot{\eta}_1} & \dots & L_{\dot{\eta}_1} \\ N_{\dot{\eta}_1} & \dots & N_{\dot{\eta}_1} \end{bmatrix} \begin{bmatrix} \dot{\eta}_1 \\ \vdots \\ \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} Y_{\eta_1} & \dots & Y_{\eta_1} \\ L_{\eta_1} & \dots & L_{\eta_1} \\ N_{\eta_1} & \dots & N_{\eta_1} \end{bmatrix} \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix} \end{aligned} \quad (10.206)$$

Furthermore, the conversions from coefficient to derivative can be shown to be as follows. Recall that $Q_\infty = \frac{1}{2}\rho_\infty \bar{v}_\infty^2$ is the dynamic pressure with trimmed airspeed $\bar{v}_\infty = \sqrt{u^2 + v^2 + w^2}$.

| \bullet | X_\bullet | Z_\bullet | M_\bullet |
|----------------|--|--|---|
| u | $\frac{Q_\infty S_w}{m\bar{v}_\infty} C_{X_u}$ | $\frac{Q_\infty S_w}{m\bar{v}_\infty} C_{Z_u}$ | $\frac{Q_\infty S_w \bar{c}_w}{I_{yy}\bar{v}_\infty} C_{m_u}$ |
| α | $\frac{Q_\infty S_w}{m} C_{X_\alpha}$ | $\frac{Q_\infty S_w}{m} C_{Z_\alpha}$ | $\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_\alpha}$ |
| q | $\frac{Q_\infty S_w \bar{c}_w}{2m\bar{v}_\infty} C_{Z_q}$ | $\frac{Q_\infty S_w \bar{c}_w}{2m\bar{v}_\infty} C_{Z_q}$ | $\frac{Q_\infty S_w \bar{c}_w^2}{2I_{yy}\bar{v}_\infty} C_{m_q}$ |
| $\dot{\alpha}$ | $\frac{Q_\infty S_w \bar{c}_w}{2m\bar{v}_\infty} C_{X_{\dot{\alpha}}}$ | $\frac{Q_\infty S_w \bar{c}_w}{2m\bar{v}_\infty} C_{Z_{\dot{\alpha}}}$ | $\frac{Q_\infty S_w \bar{c}_w^2}{2I_{yy}\bar{v}_\infty} C_{m_{\dot{\alpha}}}$ |
| δ_e | $\frac{Q_\infty S_w}{m} C_{X_{\delta_e}}$ | $\frac{Q_\infty S_w}{m} C_{Z_{\delta_e}}$ | $\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_e}}$ |
| δ_t | $\frac{Q_\infty S_w}{m} C_{X_{\delta_t}}$ | $\frac{Q_\infty S_w}{m} C_{Z_{\delta_t}}$ | $\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_t}}$ |
| η_i | $\frac{Q_\infty S_w}{m} C_{X_{\eta_i}}$ | $\frac{Q_\infty S_w}{m} C_{Z_{\eta_i}}$ | $\frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_{\eta_i}}$ |
| $\dot{\eta}_i$ | $\frac{Q_\infty S_w}{m\bar{v}_\infty} C_{X_{\dot{\eta}_i}}$ | $\frac{Q_\infty S_w}{m\bar{v}_\infty} C_{Z_{\dot{\eta}_i}}$ | $\frac{Q_\infty S_w \bar{c}_w}{I_{yy}\bar{v}_\infty} C_{m_{\dot{\eta}_i}}$ |

| \bullet | Y_\bullet | L_\bullet | N_\bullet |
|------------|---|--|--|
| β | $\frac{Q_\infty S_w}{m} C_{Y_\beta}$ | $\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_\beta}$ | $\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_\beta}$ |
| p | $\frac{Q_\infty S_w b_w}{2m\bar{v}_\infty} C_{Y_p}$ | $\frac{Q_\infty S_w b_w^2}{2I_{xx}\bar{v}_\infty} C_{l_p}$ | $\frac{Q_\infty S_w b_w^2}{2I_{zz}\bar{v}_\infty} C_{n_p}$ |
| r | $\frac{Q_\infty S_w b_w}{2m\bar{v}_\infty} C_{Y_r}$ | $\frac{Q_\infty S_w b_w^2}{2I_{xx}\bar{v}_\infty} C_{l_r}$ | $\frac{Q_\infty S_w b_w^2}{2I_{zz}\bar{v}_\infty} C_{n_r}$ |
| δ_a | $\frac{Q_\infty S_w}{m} C_{Y_{\delta_a}}$ | $\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\delta_a}}$ | $\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\delta_a}}$ |
| δ_r | $\frac{Q_\infty S_w}{m} C_{Y_{\delta_r}}$ | $\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\delta_r}}$ | $\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\delta_r}}$ |
| η_i | $\frac{Q_\infty S_w}{m} C_{Y_{\eta_i}}$ | $\frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\eta_i}}$ | $\frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\eta_i}}$ |

$$\left| \begin{array}{c|c|c|c} \dot{\eta}_i & \frac{Q_{\infty}S_w}{m\bar{v}_{\infty}}C_{Y_{\dot{\eta}_i}} & \frac{Q_{\infty}S_w b_w}{I_{xx}\bar{v}_{\infty}}C_{n_{\dot{\eta}_i}} & \frac{Q_{\infty}S_w b_w}{I_{zz}\bar{v}_{\infty}}C_{n_{\dot{\eta}_i}} \end{array} \right|$$

Lastly, one must also include the n vibration LTI ODEs where typically one includes some level of damping for each mode, ζ_i , whose value is typically assessed from matching analytical modeling with experimental data (usually quite low, e.g. 0.02). Thus, one has

$$\ddot{\eta}_i + 2\zeta_i\omega_i\dot{\eta}_i + \omega_i^2\eta_i = \frac{Q_i}{M_i}, \quad i = 1, \dots, n \quad (10.207)$$

where the generalized forces can be modeled as linear equations of the states, i.e.

$$\begin{aligned} Q_i = & Q_{i_0} + [Q_{i_u} \quad Q_{i_\beta} \quad Q_{i_\alpha} \quad Q_{i_p} \quad Q_{i_q} \quad Q_{i_r}] \begin{bmatrix} u \\ \beta \\ \alpha \\ p \\ q \\ r \end{bmatrix} + [Q_{i_{\delta_a}} \quad Q_{i_{\delta_e}} \quad Q_{i_{\delta_r}} \quad Q_{i_{\delta_t}}] \begin{bmatrix} \delta_a \\ \delta_e \\ \delta_r \\ \delta_t \end{bmatrix} \\ & + [Q_{i_{\dot{\eta}_1}} \quad \cdots \quad Q_{i_{\dot{\eta}_n}}] \begin{bmatrix} \dot{\eta}_1 \\ \vdots \\ \dot{\eta}_n \end{bmatrix} + [Q_{i_{\eta_1}} \quad \cdots \quad Q_{i_{\eta_n}}] \begin{bmatrix} \eta_1 \\ \vdots \\ \eta_n \end{bmatrix} \end{aligned} \quad (10.208)$$

where

$$Q_{i_{\bullet}} = Q_{\infty}S_w\bar{c}_wC_{Q_{i_{\bullet}}} \quad (10.209)$$

for $\bullet = \alpha, \beta, \delta_a, \delta_e, \delta_r, \delta_T, \eta_j$ for $j = 1, \dots, n$, and

$$Q_{i_{\bullet}} = \frac{Q_{\infty}S_w\bar{c}_w}{\bar{v}_{\infty}}C_{Q_{i_{\bullet}}} \quad (10.210)$$

for $\bullet = u, p, q, r, \dot{\eta}_j$ for $j = 1, \dots, n$.

Static-Elastic Airplane Effects

First, recall the elastic flight vehicle equations of motion in state-space form as

$$\begin{aligned} \vec{x}_{rig} &= f_{rig}(\vec{x}_{rig}, \phi, \theta) + \mathcal{A}_{rig \leftarrow rig} \vec{x}_{rig} + [\mathcal{A}_{rig \leftarrow \eta} \quad \mathcal{A}_{rig \leftarrow \dot{\eta}}] \vec{x}_{vib} + \mathcal{B}_{rig} \vec{u} \\ \vec{x}_{vib} &= \begin{bmatrix} 0_{n \times 6} \\ A_{vib \leftarrow rig} \end{bmatrix} \vec{x}_{rig} + \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ A_{vib \leftarrow \eta} & A_{vib \leftarrow \dot{\eta}} \end{bmatrix} \vec{x}_{vib} + \begin{bmatrix} 0_{n \times 4} \\ B_{vib} \end{bmatrix} \vec{u} \end{aligned} \quad (10.211)$$

To consider only the static-elastic effects, i.e., deformation equilibrium, one can set all $\ddot{\eta}_i = \dot{\eta}_i = 0 \forall i$. Then, one can solve for the **static-elastic modal coordinates**,

$$\bar{\eta} = [\bar{\eta}_1 \quad \cdots \quad \bar{\eta}_n] \quad (10.212)$$

in terms of the rigid vehicle state and control inputs.

Using the vibration equation of motion, i.e.,

$$\vec{0} = \begin{bmatrix} 0_{n \times 6} \\ A_{vib \leftarrow rig} \end{bmatrix} \vec{x}_{rig} + \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ A_{vib \leftarrow \eta} & A_{vib \leftarrow \dot{\eta}} \end{bmatrix} \begin{bmatrix} \bar{\eta} \\ \vec{0} \end{bmatrix} + \begin{bmatrix} 0_{n \times 4} \\ B_{vib} \end{bmatrix} \vec{u} \quad (10.213)$$

for which, the non-trivial portion states:

$$0 = A_{vib \leftarrow rig} \vec{x}_{rig} + A_{vib \leftarrow \eta} \bar{\eta} + B_{vib} \vec{u} \quad (10.214)$$

or, finally, the **static-elastic constraint**

$$\bar{\eta} = A_{vib \leftarrow \eta}^{-1} (A_{vib \leftarrow rig} \vec{x}_{rig} + B_{vib} \vec{u}) \quad (10.215)$$

Using the rigid-body equation of motion, i.e.,

$$\dot{\vec{x}}_{rig} = f_{rig}(\vec{x}_{rig}, \phi, \theta) + \mathcal{A}_{rig \leftarrow rig} \vec{x}_{rig} + [\mathcal{A}_{rig \leftarrow \eta} \quad \mathcal{A}_{rig \leftarrow \dot{\eta}}] \begin{bmatrix} \bar{\eta} \\ \vec{0} \end{bmatrix} + \mathcal{B}_{rig} \vec{u} \quad (10.216)$$

Finally, by back-substitution, one has for the **static-elastic rigid vehicle EOMs**

$$\begin{aligned} \dot{\vec{x}}_{rig} = & f_{rig}(\vec{x}_{rig}, \phi, \theta) \\ & + (\mathcal{A}_{rig \leftarrow rig} - \mathcal{A}_{rig \leftarrow \eta} A_{vib \leftarrow \eta}^{-1} A_{vib \leftarrow rig}) \vec{x}_{rig} \\ & + (B_{rig} - \mathcal{A}_{rig \leftarrow \eta} \mathcal{A}_{vib \leftarrow \eta}^{-1} B_{vib}) \vec{u} \end{aligned} \quad (10.217)$$

which is a process known as **residualization** of the vibration degrees-of-freedom into the new matrices of static-elastic stability and control derivatives/coefficients, i.e., the elements of

$(\mathcal{A}_{rig \leftarrow rig} - \mathcal{A}_{rig \leftarrow \eta} A_{vib \leftarrow \eta}^{-1} A_{vib \leftarrow rig})$ and $(B_{rig} - \mathcal{A}_{rig \leftarrow \eta} \mathcal{A}_{vib \leftarrow \eta}^{-1} B_{vib})$. It is important to note that, in general, these residualized static-elastic derivatives/coefficients depend on the flight conditions as these directly affect the loads on the vehicle's structure, especially the dynamic pressure. Furthermore, if the aerodynamic forces and moments are not truly linear, then one must use numerical techniques to find the static-elastic modal coordinates.

Linearized Elastic Flight Vehicle Dynamics

Furthermore, as shown previously for the rigid flight vehicle dynamics, one can linearize the elastic flight vehicle EOMs for easier analysis, simulation, and design. As $f_{rig}(\vec{x}_{rig}, \theta, \phi)$ and the Euler angle equations are the only nonlinear term of the EOMs, but as these terms are rigid vehicle specific, one can simply use a rigid flight vehicle linearization method to form the LTI state-space system as

$$\begin{bmatrix} \Delta \dot{\vec{x}}_{rig} \\ \Delta \dot{\vec{x}}_{eul} \\ \Delta \dot{\vec{x}}_{vib} \end{bmatrix} = \begin{bmatrix} A_{rig \leftarrow rig} & A_{rig \leftarrow eul} & A_{rig \leftarrow vib} \\ A_{eul \leftarrow rig} & A_{eul \leftarrow eul} & 0_{3 \times 2n} \\ A_{vib \leftarrow rig} & 0_{2n \times 3} & A_{vib \leftarrow vib} \end{bmatrix} \begin{bmatrix} \Delta \vec{x}_{rig} \\ \Delta \vec{x}_{eul} \\ \Delta \vec{x}_{vib} \end{bmatrix} + \begin{bmatrix} B_{rig} \\ 0_{3 \times 4} \\ B_{vib} \end{bmatrix} \Delta \vec{u} \quad (10.218)$$

where

$$\Delta \vec{x}_{eul} = \begin{bmatrix} \Delta \phi \\ \Delta \theta \\ \Delta \psi \end{bmatrix} \quad (10.219)$$

$$A_{vib \leftarrow vib} = \begin{bmatrix} 0_{n \times n} & I_{n \times n} \\ A_{vib \leftarrow \eta} & A_{vib \leftarrow \dot{\eta}} \end{bmatrix} \quad (10.220)$$

$A_{rig \leftarrow rig}$ has been altered from $\mathcal{A}_{rig \leftarrow rig}$ due to the linearization of $f_{rig}(\vec{x}_{rig}, \theta, \phi)$ with respect to \vec{x}_{rig} , $A_{rig \leftarrow eul}$ results from the linearization of $f_{rig}(\vec{x}_{rig}, \theta, \phi)$ with respect to \vec{x}_{eul} , and $A_{eul \leftarrow rig}$ and $A_{eul \leftarrow eul}$ result from the linearization of the supplemental Euler angle equation. The explicit computation of this linearized state matrix for the rigid vehicle about a general trim/reference flight condition will be addressed in more detail later.

Note that one can also use the reference and perturbation form for the modal coordinates as

$$\eta_i(t) = \bar{\eta}_i + \Delta\eta_i(t) \quad (10.221)$$

and for the pressure distribution as

$$\vec{P}(\vec{x}) = \bar{P}(\vec{x}) + \Delta\vec{P}(\vec{x}) \quad (10.222)$$

Thus, for the perturbation or linearized vibration equation of motion, one has

$$\Delta\ddot{\eta}_i + \omega_i^2 \Delta\eta_i = \frac{1}{M} \int_{Area} \Delta\vec{P}(\vec{x}_B) \cdot \vec{v}(\vec{x}_B) dS, \quad i = 1, \dots, n \quad (10.223)$$

Linearized Elastic Airplane Dynamics

As opposed to rigid body modeling, the linearized equations of motion for elastic airplanes typically use the fuselage body frame (subscript F) instead of the stability body frame (subscript S) for developing the vibration and dynamic-elastic coefficients. Thus, if one has developed the linearized rigid airplane in the stability frame, one must first transform the perturbed rigid body aerodynamic and propulsive forces and moments from the stability frame to the fuselage frame as

$$\begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}_F = \begin{bmatrix} \cos \bar{\alpha} & 0 & -\sin \bar{\alpha} \\ 0 & 1 & 0 \\ \sin \bar{\alpha} & 0 & \cos \bar{\alpha} \end{bmatrix} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix}_S \quad (10.224)$$

and

$$\begin{bmatrix} \Delta L \\ \Delta M \\ \Delta N \end{bmatrix}_F = \begin{bmatrix} \cos \bar{\alpha} & 0 & -\sin \bar{\alpha} \\ 0 & 1 & 0 \\ \sin \bar{\alpha} & 0 & \cos \bar{\alpha} \end{bmatrix} \begin{bmatrix} \Delta L \\ \Delta M \\ \Delta N \end{bmatrix}_S \quad (10.225)$$

Having redefined these terms, one may use the linearized equations of motion in the fuselage frame as opposed to the stability frame. However, in this case $\bar{\alpha}$ may not equal 0, thus, the linearized equations become

$$\begin{aligned} \begin{bmatrix} \Delta X \\ \Delta Y \\ \Delta Z \end{bmatrix} + g \begin{bmatrix} -\cos \bar{\theta} & 0 \\ -\sin \bar{\theta} \sin \bar{\phi} & \cos \bar{\theta} \cos \bar{\phi} \\ \sin \bar{\theta} \cos \bar{\phi} & \cos \bar{\theta} \sin \bar{\phi} \end{bmatrix} \begin{bmatrix} \theta \\ \phi \\ \psi \end{bmatrix} &= \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{v} \\ \Delta \dot{w} \end{bmatrix} \\ + \begin{bmatrix} 0 & -\bar{r} & \bar{q} \\ \bar{r} & 0 & -\bar{p} \\ -\bar{q} & \bar{p} & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta v \\ \Delta w \end{bmatrix} + \begin{bmatrix} 0 & \bar{w} & -\bar{v} \\ -\bar{w} & 0 & \bar{u} \\ \bar{v} & -\bar{u} & 0 \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta q \\ \Delta r \end{bmatrix} \end{aligned} \quad (10.226)$$

and

$$\begin{bmatrix} \Delta L \\ \Delta M \\ \Delta N \end{bmatrix} = \begin{bmatrix} 1 & 0 & -\frac{I_{xz}}{I_{xx}} \\ 0 & 1 & 0 \\ -\frac{I_{xz}}{I_{zz}} & 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta \dot{p} \\ \Delta \dot{q} \\ \Delta \dot{r} \end{bmatrix} + \begin{bmatrix} -\frac{I_{xz}}{I_{xx}} \bar{q} & -\frac{I_{xz}}{I_{xx}} \bar{p} + \frac{I_{zz}-I_{yy}}{I_{xx}} \bar{r} & \frac{I_{zz}-I_{yy}}{I_{xx}} \bar{q} \\ \frac{I_{xx}-I_{zz}}{I_{yy}} \bar{r} + 2 \frac{I_{xz}}{I_{yy}} \bar{p} & 1 & \frac{I_{xx}-I_{zz}}{I_{yy}} \bar{p} - 2 \frac{I_{xz}}{I_{yy}} \bar{r} \\ \frac{I_{yy}-I_{xx}}{I_{zz}} \bar{q} & \frac{I_{xz}}{I_{zz}} \bar{r} + \frac{I_{yy}-I_{xx}}{I_{zz}} \bar{p} & \frac{I_{xz}}{I_{zz}} \bar{q} \end{bmatrix} \begin{bmatrix} \Delta p \\ \Delta q \\ \Delta r \end{bmatrix} \quad (10.227)$$

where the perturbed forces and moments can be modeled as

$$\begin{bmatrix} \Delta X \\ \Delta Z \\ \Delta M \end{bmatrix} = \begin{bmatrix} 0 & X_{\dot{\alpha}} & 0 \\ 0 & Z_{\dot{\alpha}} & 0 \\ 0 & M_{\dot{\alpha}} & 0 \end{bmatrix} \begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \end{bmatrix} + \begin{bmatrix} X_u & X_{\alpha} & X_q \\ Z_u & Z_{\alpha} & Z_q \\ M_u & M_{\alpha} & M_q \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ Z_{\delta_e} & Z_{\delta_t} \\ M_{\delta_e} & M_{\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} + \begin{bmatrix} X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_n} \\ Z_{\dot{\eta}_1} & \cdots & Z_{\dot{\eta}_n} \\ M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_n} \end{bmatrix} \begin{bmatrix} \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} X_{\eta_1} & \cdots & X_{\eta_n} \\ Z_{\eta_1} & \cdots & Z_{\eta_n} \\ M_{\eta_1} & \cdots & M_{\eta_n} \end{bmatrix} \begin{bmatrix} \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \end{bmatrix} \quad (10.228)$$

and

$$\begin{bmatrix} \Delta Y \\ \Delta L \\ \Delta N \end{bmatrix} = \begin{bmatrix} Y_{\beta} & Y_p & Y_r \\ L_{\beta} & L_p & L_r \\ N_{\beta} & N_p & N_r \end{bmatrix} \begin{bmatrix} \Delta \beta \\ \Delta p \\ \Delta r \end{bmatrix} + \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ L_{\delta_a} & L_{\delta_r} \\ N_{\delta_a} & N_{\delta_r} \end{bmatrix} \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_r \end{bmatrix} + \begin{bmatrix} Y_{\dot{\eta}_1} & \cdots & Y_{\dot{\eta}_n} \\ L_{\dot{\eta}_1} & \cdots & L_{\dot{\eta}_n} \\ N_{\dot{\eta}_1} & \cdots & N_{\dot{\eta}_n} \end{bmatrix} \begin{bmatrix} \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \end{bmatrix} + \begin{bmatrix} Y_{\eta_1} & \cdots & Y_{\eta_n} \\ L_{\eta_1} & \cdots & L_{\eta_n} \\ N_{\eta_1} & \cdots & N_{\eta_n} \end{bmatrix} \begin{bmatrix} \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \end{bmatrix} \quad (10.229)$$

Note that alternatively one may also use the angle of attack and sideslip angle to make the substitutions

$$\Delta w = \bar{u} \Delta \alpha \quad (10.230)$$

and

$$\Delta v = \bar{v}_{\infty} \Delta \beta \quad (10.231)$$

under the small angle approximation and no-wind assumption. Furthermore, if $\bar{v} = \bar{\phi} = \bar{p} = \bar{q} = \bar{r} = 0$, then one can decouple the dynamics into the longitudinal and lateral-directional.

Lastly, one must also include the linearized vibration equations. However, as these are already linearly modeled, one can simply write

$$\Delta \ddot{\eta}_i + 2\zeta_i \omega_i \Delta \dot{\eta}_i + \omega_i^2 \Delta \eta_i = \frac{\Delta Q_i}{M_i}, \quad i = 1, \dots, n \quad (10.232)$$

where the generalized forces can be modeled as linear equations of the states, i.e.

$$\begin{aligned} \Delta Q_i = & [Q_{i_u} \quad Q_{i_\beta} \quad Q_{i_\alpha} \quad Q_{i_p} \quad Q_{i_q} \quad Q_{i_r}] \begin{bmatrix} \Delta u \\ \Delta \beta \\ \Delta \alpha \\ \Delta p \\ \Delta q \\ \Delta r \end{bmatrix} + [Q_{i_{\delta_a}} \quad Q_{i_{\delta_e}} \quad Q_{i_{\delta_r}} \quad Q_{i_{\delta_t}}] \begin{bmatrix} \Delta \delta_a \\ \Delta \delta_e \\ \Delta \delta_r \\ \Delta \delta_t \end{bmatrix} \\ & + [Q_{i_{\dot{\eta}_1}} \quad \cdots \quad Q_{i_{\dot{\eta}_n}}] \begin{bmatrix} \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \end{bmatrix} + [Q_{i_{\eta_1}} \quad \cdots \quad Q_{i_{\eta_n}}] \begin{bmatrix} \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \end{bmatrix} \end{aligned} \quad (10.233)$$

For an explicit example of a linearized elastic flight vehicle equation of motion, consider the simpler case where the trim flight condition is straight-and-level flight and the longitudinal and lateral-directional EOMs can be decoupled. Furthermore, assume that the vibration modes can also be decoupled between the longitudinal and lateral-directional and $\dot{\alpha}$ derivatives are zero. Then, the longitudinal rigid airplane LTI state-space model

$$\begin{bmatrix} \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & X_q & -g \\ \frac{Z_u}{\bar{u}} & \frac{Z_\alpha}{\bar{u}} & 1 + \frac{Z_q}{\bar{u}} & 0 \\ M_u & M_\alpha & M_q & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ \frac{Z_{\delta_e}}{\bar{u}} & \frac{Z_{\delta_t}}{\bar{u}} \\ M_{\delta_e} & M_{\delta_t} \\ 0 & 0 \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (10.234)$$

which models the following portions of Equation 10.218 as

$$\begin{bmatrix} \Delta \dot{\vec{x}}_{rig} \\ \Delta \dot{\vec{x}}_{eul} \end{bmatrix} = \begin{bmatrix} A_{rig \leftarrow rig} & A_{rig \leftarrow eul} \\ A_{eul \leftarrow rig} & A_{eul \leftarrow eul} \end{bmatrix} \begin{bmatrix} \Delta \vec{x}_{rig} \\ \Delta \vec{x}_{eul} \end{bmatrix} + \begin{bmatrix} B_{rig} \\ 0 \end{bmatrix} \Delta \vec{u} \quad (10.235)$$

Note that here $\Delta \alpha$ has been used in place of Δw . Then, one may form the other matrices as

$$A_{rig \leftarrow vib} = \begin{bmatrix} X_{\eta_1} & \cdots & X_{\eta_n} & X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_n} \\ \frac{Z_{\eta_1}}{\bar{u}} & \cdots & \frac{Z_{\eta_n}}{\bar{u}} & \frac{Z_{\dot{\eta}_1}}{\bar{u}} & \cdots & \frac{Z_{\dot{\eta}_n}}{\bar{u}} \\ M_{\eta_1} & \cdots & M_{\eta_n} & M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_n} \end{bmatrix} \quad (10.236)$$

and defining the **elastic stability and control derivative** for the state or input \bullet as

$$\Xi_{i\bullet} = \frac{Q_{i\bullet}}{\mathcal{M}_i} \quad (10.237)$$

one can write the vibration state and input sub-matrices as

$$A_{vib \leftarrow rig} = \begin{bmatrix} \Xi_{1_u} & \Xi_{1_\alpha} & \Xi_{1_q} \\ \vdots & \vdots & \vdots \\ \Xi_{n_u} & \Xi_{n_\alpha} & \Xi_{n_q} \end{bmatrix} \quad (10.238)$$

$$A_{vib \leftarrow \eta} = \begin{bmatrix} \Xi_{1\eta_1} & \cdots & \Xi_{1\eta_n} \\ \vdots & \ddots & \vdots \\ \Xi_{n\eta_1} & \cdots & \Xi_{n\eta_n} \end{bmatrix} - \Omega^2 \quad (10.239)$$

$$A_{vib \leftarrow \dot{\eta}} = \begin{bmatrix} \Xi_{1\eta_1} & \cdots & \Xi_{1\dot{\eta}_n} \\ \vdots & \ddots & \vdots \\ \Xi_{n\eta_1} & \cdots & \Xi_{n\dot{\eta}_n} \end{bmatrix} - 2\Omega_\zeta \quad (10.240)$$

and

$$B_{vib} = \begin{bmatrix} \Xi_{1\delta_e} & \Xi_{1\delta_t} \\ \vdots & \vdots \\ \Xi_{n\delta_e} & \Xi_{n\delta_t} \end{bmatrix} \quad (10.241)$$

Thus, in the end, one has

$$\begin{bmatrix} \Delta u \\ \Delta \dot{u} \\ \Delta \dot{\alpha} \\ \Delta \dot{q} \\ \Delta \dot{\theta} \\ \Delta \dot{\eta}_1 \\ \vdots \\ \Delta \dot{\eta}_n \\ \Delta \ddot{\eta}_1 \\ \vdots \\ \Delta \ddot{\eta}_n \end{bmatrix} = \begin{bmatrix} X_u & X_\alpha & X_q & -g & X_{\eta_1} & \cdots & X_{\eta_n} & X_{\dot{\eta}_1} & \cdots & X_{\dot{\eta}_n} \\ \frac{Z_u}{\dot{u}} & \frac{Z_\alpha}{\dot{u}} & 1 + \frac{Z_q}{\dot{u}} & 0 & \frac{Z_{\eta_1}}{\dot{u}} & \cdots & \frac{Z_{\eta_n}}{\dot{u}} & \frac{Z_{\dot{\eta}_1}}{\dot{u}} & \cdots & \frac{Z_{\dot{\eta}_n}}{\dot{u}} \\ M_u & M_\alpha & M_q & 0 & M_{\eta_1} & \cdots & M_{\eta_n} & M_{\dot{\eta}_1} & \cdots & M_{\dot{\eta}_n} \\ 0 & 0 & 1 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 1 \\ \Xi_{1u} & \Xi_{1\alpha} & \Xi_{1q} & 0 & \Xi_{1\eta_1} - \omega_1^2 & \cdots & \Xi_{1\eta_n} & \Xi_{1\dot{\eta}_1} - 2\zeta_1\omega_1 & \cdots & \Xi_{1\dot{\eta}_n} \\ \vdots & \vdots & \vdots & 0 & \vdots & \ddots & \vdots & \vdots & \ddots & \vdots \\ \Xi_{n_u} & \Xi_{n_\alpha} & \Xi_{n_q} & 0 & \Xi_{n\eta_1} & \cdots & \Xi_{n\eta_n} - \omega_n^2 & \Xi_{n\dot{\eta}_1} & \cdots & \Xi_{n\dot{\eta}_n} - 2\zeta_n\omega_n \end{bmatrix} \begin{bmatrix} \Delta u \\ \Delta \alpha \\ \Delta q \\ \Delta \theta \\ \Delta \eta_1 \\ \vdots \\ \Delta \eta_n \\ \Delta \ddot{\eta}_1 \\ \vdots \\ \Delta \ddot{\eta}_n \end{bmatrix} + \begin{bmatrix} X_{\delta_e} & X_{\delta_t} \\ \frac{Z_{\delta_e}}{\dot{u}} & \frac{Z_{\delta_t}}{\dot{u}} \\ M_{\delta_e} & M_{\delta_t} \\ \vec{0}_{n+1} & \vec{0}_{n+1} \\ \Xi_{1\delta_e} & \Xi_{1\delta_t} \\ \vdots & \vdots \\ \Xi_{n\delta_e} & \Xi_{n\delta_t} \end{bmatrix} \begin{bmatrix} \Delta \delta_e \\ \Delta \delta_t \end{bmatrix} \quad (10.242)$$

10.5 Elastic Satellite Dynamics

Introductory Aerospace Vehicle Control Systems

11.1 Aerospace Actuation Systems

Electromechanical Actuators

The actuation system is the primary limiting factor for the inner-loop control systems of flight vehicles. The control surfaces for modern aircraft are fly-by-wire (FBW), i.e., the vehicles convert electrical signals into mechanical movements, either linear or rotary in order to control the vehicle. These fly-by-wire actuators can be electro-mechanical, electro-hydraulic, or electro-hydrostatic control actuators. For semi-automatic or automatic control to occur for these, the actuation system for airplanes must use FBW systems provide a feedback system for the automa

Electric Motor Actuator Model

The electro-mechanical dynamical system for a brushless direct current (BLDC) electric motor can be modeled using Kirchoff's second law of circuits and Euler's second law of motion as following two equations:

$$\begin{aligned} L_m \frac{di_m}{dt}(t) &= v_m(t) - R_m i_m(t) - v_{emf} \\ J_t \frac{d\omega_m}{dt}(t) &= \sum M = M_{emf,m} - M_{D,m} - M_{f,m} \end{aligned} \quad (11.1)$$

where $i_m(t)$ is the motor current, L_m is the inductance of the motor coils, $v_m(t)$ is the motor voltage, R_m is the resistance of the motor coils, J_t is the *axial* polar moment of inertia, M_m is the sum of moments on the motor, including the electromotion, $M_{emf,m}$, drag loading, $M_{D,m}$, motor friction loading, $M_{f,m}$, and v_{emf} is the electromotive force which is proportional to the motor angular velocity

$$v_{emf} = k_{emf} \omega_m \quad (11.2)$$

where k_{emf} is the electromotive constant. Lorentz's law which describes the electromotive force upon a coil in a magnetic field results in the produced moment being proportional to the motor current, i.e.

$$M_{emf,m} = k_m i_m(t) \quad (11.3)$$

where k_m is the armature constant. Assuming the motor friction is negligible, i.e. $M_{f,m} \approx 0$, one has a second-order nonlinear dynamics equation

$$\begin{aligned} \frac{di_m}{dt}(t) &= \frac{1}{L_m} v_m(t) - \frac{R_m}{L_m} i_m(t) - \frac{k_{emf}}{L_m} \omega_m(t) \\ \frac{d\omega_m}{dt}(t) &= \frac{k_m}{J_t} i_m(t) - \frac{1}{J_t} M_{D,m} \end{aligned} \quad (11.4)$$

The gearbox relationship from the motor to the propeller can be modeled by two equations:

$$\begin{aligned} \omega_m &= r_g \omega_p \\ M_m &= \frac{1}{\eta_g r_g} M_p \end{aligned} \quad (11.5)$$

where r_g is the gearbox reduction ratio, η_g is the gearbox efficiency (typically 80 – 90%), ω_m is the angular velocity of the motor, ω_p is the angular velocity of the propeller, M_m is the load moment that the motor experiences, and M_p is the load moment that the propeller experiences. Note that ideally the power must be conserved as

$$M_m \omega_m = M_p \omega_p \quad (11.6)$$

but the efficiency comes into play due to the heat loss. Lastly, it should be noted that the total polar moment of inertia experienced by the motor, J_t , combines the motor shaft moment of inertia, J_m , with the propeller moment of inertia, J_p , through the gearbox reduction ratio as

$$J_t = J_m + \frac{J_p}{r_g^2} \quad (11.7)$$

By substitution of the second propeller equation into the first gearbox equation, one has

$$M_{D,p} = \frac{C_{dm}}{r_g^2} \omega_m^2 \quad (11.8)$$

Then, by substitution into the second gearbox equation, one has

$$M_{D,m} = \frac{C_{dm}}{\eta_g r_g^3} \omega_m^2 \quad (11.9)$$

Finally, substituting $M_{D,m}$ into the motor dynamics equation, an **geared electric motor** can be modeled as the nonlinear dynamical system

$$\begin{bmatrix} \frac{di_m}{dt}(t) \\ \frac{d\omega_m}{dt}(t) \end{bmatrix} = \begin{bmatrix} -\frac{R_m}{L_m} i_m(t) - \frac{k_{emf}}{L_m} \omega_m(t) + \frac{1}{L_m} u(t) \\ \frac{k_m}{J_t} i_m(t) - \frac{C_{dm}}{\eta_g r_g^3 J_t} \omega_m^2 \end{bmatrix} \quad (11.10)$$

$$y(t) = \omega_m(t)$$

where $[i_m(t) \ \omega_m(t)]^T$ is the state vector, motor voltage input, $v_m(t)$, and $\omega_m(t)$ is the output variable which is typically controlled using a SISO LTI controller.

Momentum Exchange Devices

Alternatively to expelling mass using thrusters, one may use a **momentum exchange devices (MED)**, i.e. an axisymmetric rotational device, to affect the total angular momentum of the vehicle. MEDs fundamentally contains a **flywheel**, also known as an **inertia wheel**, which can be used to store or transfer angular momentum to affect the vehicle's total angular momentum. **Momentum wheels** are flywheels designed to operate with some nonzero momentum and **reaction wheels** are flywheels designed to operate with zero momentum. These flywheels are typically connected rigidly to the satellite, but their wheels are driven by electric motors and rotate more or less independently, depending on the type of gyro.

For the satellite attitude dynamics of flywheel-equipped satellites, suppose one has a satellite of mass m_0 with center of mass G and N flywheels with angular velocities, $\omega_1, \dots, \omega_N$, masses m_1, \dots, m_N , and center of masses G_1, \dots, G_N . The mass of the entire system is

$$m = m_0 + \sum_{i=1}^N m_i \quad (11.11)$$

and the angular momentum of the entire system about the center of mass is

$$\vec{H}_G = \vec{H}_{G,v} + \vec{H}_w \quad (11.12)$$

where $\vec{H}_{G,v}$ is the total angular momentum of the rigid body comprising the satellite and all the flywheel masses concentrated at their centers of mass with common angular momentum, $\vec{\omega}$, and \vec{H}_w is the net angular momentum of the N flywheels about each of their center of masses.

Thus,

$$\vec{H}_{G,v} = I_{G,v} \vec{\omega} = \left(I_{G,0} + \sum_{i=1}^N I_{G,m_i} \right) \vec{\omega} \quad (11.13)$$

where $I_{G,0}$ is the inertia tensor of the satellite body and I_{G,m_i} is the inertia tensor of the condensed point-mass of the i^{th} flywheel located at (x_i, y_i, z_i) in the body-fixed frame, i.e.

$$I_{G,m_i} = \begin{bmatrix} m_i(x_i^2 + z_i^2) & -m_i x_i y_i & -m_i x_i z_i \\ -m_i x_i y_i & m_i(y_i^2 + z_i^2) & -m_i y_i z_i \\ -m_i x_i z_i & -m_i y_i z_i & m_i(x_i^2 + y_i^2) \end{bmatrix} \quad (11.14)$$

Both of these are constant with respect to the body frame. Also, for the flywheels' net angular momentum

$$\vec{H}_w = \sum_{i=1}^N \vec{H}_{G_i} = \sum_{i=1}^N I_{G_i} \vec{\omega}_i \quad (11.15)$$

where \vec{H}_{G_i} is the angular momentum for the i^{th} flywheel and I_{G_i} is the inertia tensor for the i^{th} flywheel about its center mass, G_i , along axes parallel to the body-fixed frame. Notably, I_{G_i} may vary with time if the flywheels' can pivot on gimbals.

Altogether, for the wheel-equipped satellite attitude dynamics, one has

$$\vec{M}_{G,a} + \vec{M}_{G,p} + \vec{M}_{G,g} = \dot{\vec{H}}_{G,v} + [\vec{\omega}] \times \vec{H}_{G,v} + \dot{\vec{H}}_w + [\vec{\omega}] \times \vec{H}_w \quad (11.16)$$

And for torque-free motion, one has from the conservation of angular momentum about the vehicle's center of mass

$$\dot{\vec{H}}_{G,v} + [\vec{\omega}] \times \vec{H}_{G,v} + \dot{\vec{H}}_w + [\vec{\omega}] \times \vec{H}_w = 0 \quad (11.17)$$

or

$$\vec{H}_{G,v} + \vec{H}_w = \mathbf{c} \quad (11.18)$$

where \mathbf{c} is a constant.

A **control moment gyro (CMG)**, also known as a **gyrotorquer**, uses a single- or double-gimballed momentum wheel spinning at a constant rate. These gyros spin at several thousand revolutions per minute (rpm). At each gimbal axis are placed electric motors called **torquers** which exert moments on the satellite when tilting the gimbal normal to the gimbal axis.

Assuming the angular momentum is directed totally along the spin axis, one has for the i^{th} CMG

$$\vec{H}_{G_i} = I_{G_i} \vec{\omega}_i \quad (11.19)$$

where $\vec{\omega}_w$ is the absolute angular velocity of the spinning flywheel, i.e.

$$\vec{\omega}_i = \vec{\omega} + \vec{\omega}_{p,i} + \vec{\omega}_{n,i} + \vec{\omega}_{s,i} \quad (11.20)$$

where $\vec{\omega}$ is the angular velocity of the vehicle to which the CMG is attached, $\vec{\omega}_{p,i}$ is the gyro's precession rate, $\vec{\omega}_{n,i}$ is the gyro's nutation rate, and $\vec{\omega}_{s,i}$ is the gyro's spin rate. However, CMGs are designed such that the $\|\vec{\omega}_{s,i}\|_2$ is three or more orders of magnitude greater than the other three components.

Thus, for the axisymmetric flywheel in the CMG-fixed frame C , one has the simplified

$$\vec{H}_{G_i,C}^{(i)} = I_{G_i} \vec{\omega}_{s,i} = \begin{bmatrix} 0 \\ 0 \\ I_{3,i} \omega_{s,i} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ H_i \end{bmatrix} \quad (11.21)$$

Then, relative to the body-fixed frame of the satellite, one has

$$\vec{H}_{G_i,B}^{(i)} = \begin{bmatrix} H_i \sin \theta_i \cos \phi_i \\ H_i \sin \theta_i \sin \phi_i \\ H_i \cos \theta_i \end{bmatrix} \quad (11.22)$$

and for the attitude dynamics of a satellite with N CMGs, one has

$$\sum \vec{M}_G = \dot{\vec{H}}_{G,v} + [\vec{\omega}] \times \vec{H}_{G,v} + \sum_{i=1}^N \dot{\vec{H}}_{G_i} + [\vec{\omega}] \times \vec{H}_i \quad (11.23)$$

$$\begin{aligned} \sum \vec{M}_G &= \begin{bmatrix} I_1 p \\ I_2 q \\ I_3 r \end{bmatrix} + \begin{bmatrix} (I_3 - I_2) qr \\ (I_1 - I_3) pr \\ (I_2 - I_1) pq \end{bmatrix} \\ &+ \sum_{i=1}^N \left(\begin{bmatrix} \sin \theta_i \cos \phi_i \dot{H}_i + H_i \cos \theta_i \cos \phi_i \dot{\theta}_i - H_i \sin \theta_i \sin \phi_i \dot{\phi}_i \\ \sin \theta_i \sin \phi_i \dot{H}_i + H_i \cos \theta_i \sin \phi_i \dot{\theta}_i - H_i \sin \theta_i \cos \phi_i \dot{\phi}_i \\ \dot{H}_i \cos \theta_i - H_i \sin \theta_i \dot{\theta}_i \end{bmatrix} + \begin{bmatrix} q \cos \theta_i - r \sin \theta_i \sin \phi_i \\ r \sin \theta_i \cos \phi_i - p \cos \theta_i \\ p \sin \theta_i \sin \phi_i - q \sin \theta_i \cos \phi_i \end{bmatrix} H_i \right) \end{aligned} \quad (11.24)$$

$$\begin{aligned} \sum \vec{M}_G = & \begin{bmatrix} I_1 p + (I_3 - I_2)qr \\ I_2 q + (I_1 - I_3)pr \\ I_3 r + (I_2 - I_1)pq \end{bmatrix} \\ & + \sum_{i=1}^N \left(\begin{bmatrix} \sin \theta_i \cos \phi_i \dot{H}_i + H_i \cos \theta_i \cos \phi_i \dot{\phi}_i - H_i \sin \theta_i \sin \phi_i \dot{\phi}_i + (H_i \cos \theta_i)q - (H_i \sin \theta_i \sin \phi_i)r \\ \sin \theta_i \sin \phi_i \dot{H}_i + H_i \cos \theta_i \sin \phi_i \dot{\phi}_i - H_i \sin \theta_i \cos \phi_i \dot{\phi}_i + (H_i \sin \theta_i \cos \phi_i)r - (H_i \cos \theta_i)p \\ H_i \cos \theta_i - H_i \sin \theta_i \dot{\phi}_i + p(H_i \sin \theta_i \sin \phi_i) - (H_i \sin \theta_i \cos \phi_i)q \end{bmatrix} \right) \end{aligned} \quad (11.25)$$

where the CMGs noticeably add to the coupling angular rates based on the CMG angles, θ_i and ϕ_i . Thus, using CMGs for attitude control is much more complex and consideration is left for future

Magnetic Torquers

Thruster Actuators

A thruster

A **thruster pair** is used to apply a pure moment to the spinning satellite as opposed to a single thruster designed for orbital maneuvers. Thruster pairs are typically small rockets and are mounted in principal planes, i.e. planes normal to principal axes, passing through the center of mass of the satellite. Thruster pairs are also accompanied by another thruster pair of rockets pointing in the opposite direction to exert a moment in the opposite sense. A thruster pair allows one to directly change the angular momentum impulsively, i.e.

$$\Delta \vec{H}_G = \int_0^{\Delta t} \vec{M}_G dt \quad (11.26)$$

where Δt is the time length of the impulse and $\Delta \vec{H}_G$ is typically applied normal to the spin axis, i.e. the nominal axis for H_G .

Suppose the thruster pairs are positioned at some \vec{r}_T and $-\vec{r}_T$ to the center of mass and if \vec{T} is their individual thrust, one has for the change in angular momentum after some brief time interval

$$\Delta \vec{H}_G = \int_0^{\Delta t} [\vec{r}_T] \times \vec{T} \Delta t + [-\vec{r}_T] \times (-\vec{T}) \Delta t \quad (11.27)$$

or

$$\Delta \vec{H}_G = 2[\vec{r}_T] \times \vec{T} \Delta t \quad (11.28)$$

This $\Delta \vec{H}_G$ is analogous to the Δv of orbital maneuvers.

References

For more information, please refer to the following

- Curtis, H. D., “10.7 Attitude Control Thrusters,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 506-509
- Nelson, R. C., “8.3 Control Surface Actuator,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 288-292

- Schmidt, D. K., “12.6 Path Guidance - ILS Couplers and VOR Homing,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 753-771
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.6 Autopilots,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 322-344

11.2 Aerospace Guidance and Control Systems

Guidance and Control Systems

For MIMO systems, such as aerospace vehicles, one may use state-space methods to develop suitable controllers, typically with optimal, adaptive, and/or robust control considerations. However, for many systems with only a few number of inputs and/or outputs, the different outputs that one desires to control may have responses on much different time scales which allow SISO methods to be applied at these different time scales through cascade control.

For example, consider the LTI systems represented by the following transfer functions:

$$G_{fast}(s) = \frac{y_{fast}(s)}{u(s)} \quad (11.29)$$

and

$$G_{slow}(s) = \frac{y_{slow}(s)}{u(s)} \quad (11.30)$$

Next, assuming that a change in y_{fast} naturally causes a change in y_{slow} with the relationship, i.e.

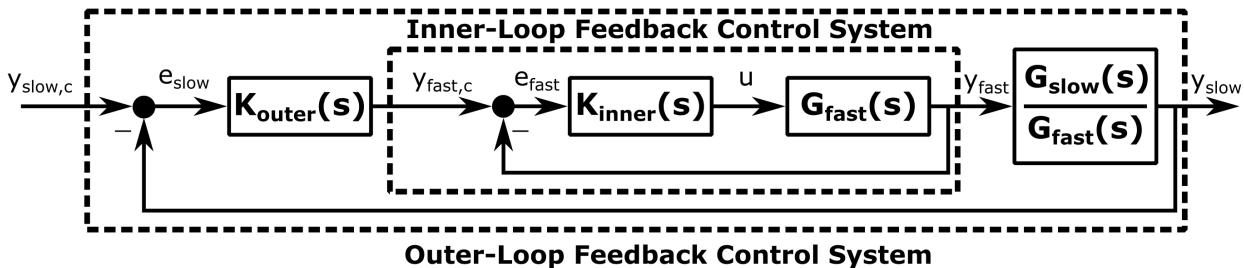
$$\frac{y_{slow}(s)}{y_{fast}(s)} = \frac{G_{slow}(s)}{G_{fast}(s)} \quad (11.31)$$

one has the fact that

$$\frac{y_{slow}(s)}{u(s)} = \frac{G_{slow}(s)}{G_{fast}(s)} G_{fast}(s) \quad (11.32)$$

Then, by assuming that y_{fast} and y_{slow} are related to one another through at least one integration, one knows that the denominator of $\frac{G_{slow}(s)}{G_{fast}(s)}$ is at least order one higher in s than the numerator.

With this in mind, consider the following block diagram which uses two linked feedback control systems, i.e. an outer feedback control system and an inner feedback control system.



where this technique is known as **cascade control**, **nested-loop control**, or **inner-outer loop control**. Although this control design may seem complicated, with proper loop-shaping, this type of feedback control system can be simpler to design than a MIMO system while also being robust with SISO LTI system stability margins.

For cascade control, one designs $K_{inner}(s)$ such that y_{fast} tracks $y_{fast,c}$ for all frequencies up to $\omega_{c,inner}$. Then, note that this makes the inner-loop feedback control system have a transfer function from $y_{fast,c}$ to y_{fast} as

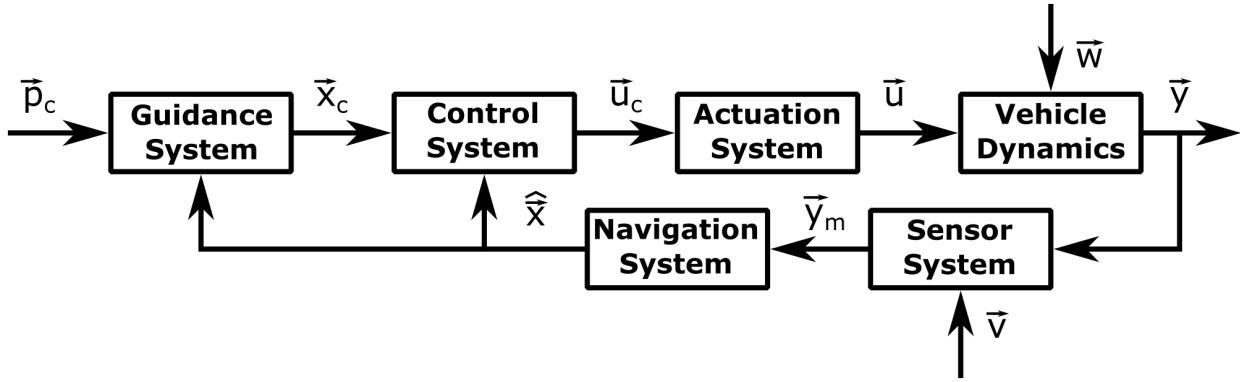
$$\frac{y_{fast}(s)}{y_{fast,c}(s)} \approx \frac{\omega_{c,inner}}{s + \omega_{c,inner}} \quad (11.33)$$

which for $\omega < \omega_{c,inner}$, this transfer function will be approximately unity below the inner-loop crossover frequency $\omega_{c,inner}$, i.e. the inner-loop bandwidth. This fact makes the preliminary design of the outer-loop much easier where assuming Equation 11.33, one can design $K_{outer}(s)$ such that y_{slow} tracks $y_{slow,c}$ for all frequencies up to $\omega_{c,outer}$ which takes advantage of the simpler form of $\frac{G_{slow}(s)}{G_{fast}(s)}$ as opposed to $G_{slow}(s)u(s)$ coupled with $G_{fast}(s)u(s)$. This two-step approach thus can be seen as much simpler than a one-step coupled design approach.

However, this cascade control method requires that there is sufficient **crossover frequency separation** between the inner- and outer-loops, i.e. $\omega_{c,outer} < \omega_{c,inner}$ by some amount such that the assumption of Equation 11.33 applies. For systems where there is already a natural frequency separation between $G_{outer}(s)$ and $G_{inner}(s)$, this can be physically realizable, but must be conserved when performing the loop-shaping of the inner- and outer-loops. In particular, standard loop-shaping of $L_{inner} = G_{inner}K_{inner}$ and $L_{outer} = G_{outer}K_{outer}$ will set $|L_{inner}|$ and $|L_{outer}|$ large for $\omega < \omega_{c,inner}$ and $\omega < \omega_{c,outer}$, respectively, and small for $\omega > \omega_{c,inner}$ and $\omega > \omega_{c,outer}$, respectively. Thus, by designing $\omega_{c,outer} < \omega_{c,inner}$ closing the outer-loop around the inner-loop will only slightly modify $|L_{inner}|$ for $\omega < \omega_{c,inner}$ where the magnitude was already large. In a similar manner, this type of logic can also be used extended to demonstrating that cascade design will have little to no affect on the inner- and outer-loop bandwidths and stability margins using loop-shaping. It should also be noted that in this way, multiple inner-loops can be cascaded which is typical for vehicle feedback control systems.

Guidance, Navigation, and Control Systems

The entire control system for vehicles are also known as **guidance, navigation, and control (GNC)** systems which refers to the three traditional sub-systems used to control the motion of vehicles. The block diagram of a typical GNC system is shown as follows.



where the vehicle state vector, \vec{x} , typically includes the position, velocity, and attitude of the vehicle. Here, the vehicle's state, \vec{x} , evolves in time according to its natural dynamics, the control inputs, \vec{u} , and any disturbances, \vec{w} , from the environment. It should be noted that for manual control of aerospace vehicles, the guidance and control systems can be considered as the human operator and not a computer system.

At the base of guidance, navigation, and control systems is the necessity of the pilot or autopilot to use a **sensor system** to sense the vehicle's state vector. In addition, a corresponding **navigation system** is typically necessary to process, i.e. “filter,” the raw signals obtained by the sensor system. Furthermore, when only certain aspects of the vehicle dynamics, i.e. the outputs \vec{y} , are measurable by the sensor system, the navigation system estimates the state vector, $\hat{\vec{x}}$, that is required for the guidance and control systems to operate. Thus, **navigation** can be defined as **vehicle (self-)state estimation**. This part of the textbook introduces some sensor systems for direct attitude determination while later parts of this textbook further develop the theory and design of full navigation systems for aerospace vehicles.

Given the current vehicle state vector, the **guidance system** determines the commanded vehicle state vector, \vec{x}_c , for following some reference path or trajectory, \vec{p}_{traj} , e.g. determines a commanded velocity, attitude, and acceleration. Notably, this system is often simply controlled by a human operator and does not use automatic control algorithms. However, as the input to the operator is the available human sensory information and/or an information display from sensors, as it would be for electronic information from the navigation system, the operator is still using the basic logic of GNC reasoning. The **control system** achieves the commanded state, \vec{x}_c , while also maintaining vehicle dynamic stability and typically uses either automatic or semi-automatic control. Furthermore, as the control system include the direct manipulation of the forces applied to the vehicle through some sort of physical phenomena, one should also consider the effects of any **actuation system** on the GNC system. Coupled with the vehicle dynamics, these actuator dynamics play a vital role in the control system design for vehicles.

The vast majority of aerospace vehicles use a cascade control scheme for guidance and control which takes advantage of the physics-based relationships between position, velocity, and acceleration as well as angular versus linear states, both of which naturally have frequency separation required for cascade control. As such the different control loops for vehicles have been given different names: planning, guidance, and control. From a feedback control perspective for aerospace vehicles, planning roughly corresponds to position or trajectory feedback control, guidance to velocity feedback control, and control to attitude feedback control directly through the actuation system. The actuation system is the primary limiting factor for the inner-loop

control systems of aerospace vehicles. It should also be noted that some aerospace vehicles may also have a SAS to alter the modal characteristics of the vehicle. In many cases, the reference trajectory may simply be commands for speed, altitude, heading. More generally, the planned trajectory may be a line in space and the system may be some form of **line-following guidance law**. Multiple lines may connect together to form a sequence of **waypoints**. Then, the switching from each line connecting waypoints is performed by the planning system of the aerospace vehicle as it assesses the current position estimate of the aerospace vehicle and often uses some success and feasibility criteria checks for reaching each waypoint.

The reference path or trajectory, \vec{p}_c , determined using **planning** at the mission level and the path level. Using the most general definition of a **plan**, i.e. a task for a vehicle that transports a payload, a vehicle must travel from current location to one (or more) designated target(s). The determination of the target is known as **mission planning** and the determination of the path to the target is known as **path planning**, also known as **trajectory planning**. As such, one can also consider the operation of autonomous aerospace vehicles at a level higher than GNC and is generally performed by a **planning system** that determines the current designated target and/or the reference path or trajectory that the vehicle should follow from its current location to the target. This stage is often completely manually derived and pre-programmed without real-time signal inputs, but autonomous planning of aerospace vehicles is also possible. Lastly, if one must reach dynamic target(s), an additional **tracking system** will be necessary to accomplish the flight plan. Furthermore, if more than one target specified, a planning system must have feedback from the navigation and target tracking system to assess when to switch from targets. Later parts of this textbook further develop the theory and design of planning and tracking systems for aerospace vehicles.

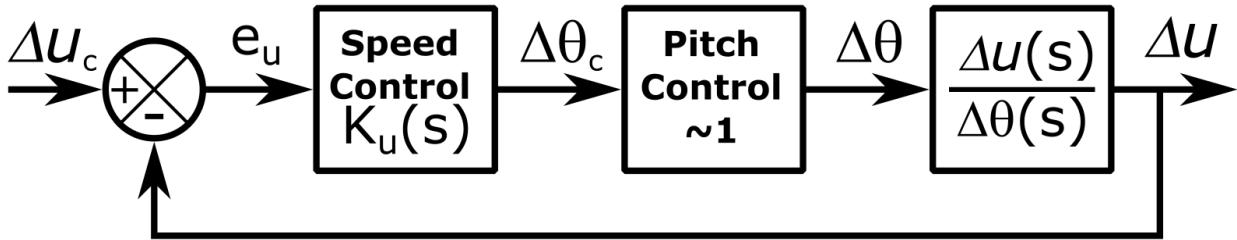
Beyond this general framework, it should be noted that sometimes the guidance and control systems are combined into one feedback control system which is possible when state-space methods are used for feedback control system design to provide faster system responses in aerospace vehicle dynamics than is possible with a traditional cascade control design. It should also be noted that when *designing* the guidance and control systems, one is typically using linearized vehicle dynamics models. Thus, aerospace vehicles often use some form of **adaptive control** which alters the guidance and control laws based on current operating conditions. A basic example of this is **gain scheduling** where the guidance and control laws or “gains” change according to the “scheduled” trim conditions. However, when employing gain scheduling, one typically requires that the transition from one scheduled trim condition to another is relatively “smooth,” i.e. the nonlinear dynamics are not significantly excited by the control inputs during the transitions between trim conditions. This is typically dealt with by designing sufficient stability robustness for each LTI controller to satisfy this requirement. As part of the gain scheduling of aircraft, one typically requires the estimation of the current flight conditions in real-time. This is often performed by an onboard sensor system known as an **air data system (ADS)** which provide measurements of a aerospace vehicle’s surrounding air mass, collectively known as the **air data**, typically quantified as the airspeed, angle of attack, sideslip angle, and perhaps the altitude and rate of climb. The airspeed, angle of attack, and sideslip angle are also known as the **air data triplet**.

Introductory Airplane Guidance Systems

Though airplanes are inherently MIMO systems, due to the standard steady-flight conditions for the majority of airplanes, the longitudinal and lateral-directional feedback control systems are typically designed separately and operated in parallel using the different control inputs for controlling the inner-loops of the

airplane. In particular, the elevator and throttle are used for the longitudinal feedback control system and the rudder and ailerons are used for the lateral-directional feedback control systems. The inner-loops in both cases are first for attitude control, i.e. pitch angle (or angle of attack or pitch rate) and roll angle, respectively, which are the fundamental attitude angles to control as they directly affect the magnitude and direction of the airplane's lift vector, respectively. Closed around the inner-loop attitude controllers are the outer-loops guidance laws for both longitudinal and lateral which follow the planned reference trajectory for which the simplest are holds on speed, altitude, and heading. For long-distance flying, one typically prescribes the line as the shortest distance from one location to another, also known as a **great circle** line, as it follows the curvature of the Earth's reference ellipsoid.

One option for the longitudinal outer-loop guidance system for an airplane is a **speed hold** which is typically employed during climbing flight under air traffic control. This guidance loop uses the speed control law, $K_u(s)$, the pitch inner-loop control system, and the transfer function from $\Delta\theta \rightarrow \Delta u$ for the outer-loop plant as shown in the following block diagram.

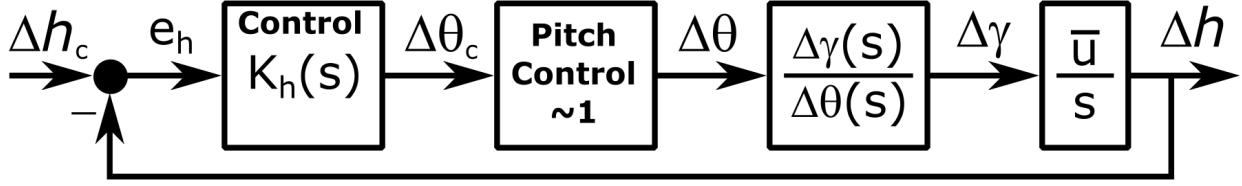


As the inner-loop of this speed hold control loop uses the elevator, one can form the outer-loop plant as

$$\frac{\Delta u(s)}{\Delta\theta(s)} = \frac{\text{num}\left(\frac{\Delta u(s)}{\Delta\delta_e(s)}\right)}{\text{num}\left(\frac{\Delta\theta(s)}{\Delta\delta_e(s)}\right)} \quad (11.34)$$

where $\text{num}(G(s))$ represents the numerator of the transfer function $G(s)$. To form this model, one may use the transfer functions $\frac{\Delta u(s)}{\Delta\delta_e(s)}$ and $\frac{\Delta\theta(s)}{\Delta\delta_e(s)}$ for the vehicle alone as long as the crossover frequency separation for the inner- and outer-loops is maintained. Typically, one also includes the pitch inner-loop control system once it has been designed. It should be noted that $K_u(s)$ or $\frac{\Delta u(s)}{\Delta\theta(s)}$ may need to be negative as a positive change in pitch $\Delta\theta$ may cause a reduction in flight velocity u .

Another option for the longitudinal outer-loop guidance system for an airplane is an **altitude hold** which is typically employed during a cruise flight condition at a specific cruise velocity specified by the throttle setting. This guidance loop uses the altitude control law, $K_h(s)$, the pitch inner-loop control system, and the transfer function from $\Delta\theta \rightarrow \Delta\gamma$ for the outer-loop plant as shown in the following block diagram.



It should be noted that the transfer function from $\Delta\theta \rightarrow \Delta\gamma$ can be calculated using the small angle approximation for the Euler angles as

$$\Delta\gamma(s) = \Delta\theta(s) - \Delta\alpha(s) \quad (11.35)$$

which provides

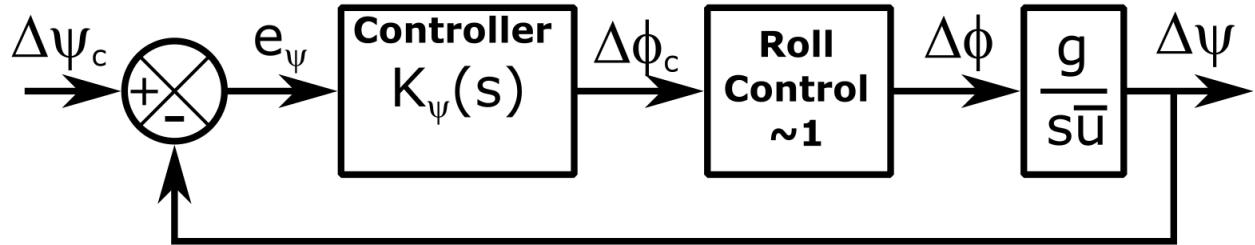
$$\frac{\Delta\gamma(s)}{\Delta\theta(s)} = 1 - \frac{\Delta\alpha(s)}{\Delta\theta(s)} \quad (11.36)$$

or

$$\frac{\Delta\gamma(s)}{\Delta\theta(s)} = 1 - \frac{\text{num}\left(\frac{\Delta\alpha(s)}{\Delta\delta_e(s)}\right)}{\text{num}\left(\frac{\Delta\theta(s)}{\Delta\delta_e(s)}\right)} \quad (11.37)$$

To form this model, one may again use the transfer functions $\frac{\alpha(s)}{\delta_e(s)}$ and $\frac{\theta(s)}{\delta_e(s)}$ for the vehicle alone as long as the crossover frequency separation for the inner- and outer-loops is maintained. Typically, one also includes the pitch inner-loop control system once it has been designed. It should also be noted that here it is required that an increase in pitch angle θ must produce a steady-state flight path angle γ , i.e. the airplane must be “on the front side of the power curve.”

One option for the lateral-directional outer-loop guidance system for an airplane is a **heading hold** which is typically employed at a specific cruise velocity. This guidance loop uses the heading control law, $K_\psi(s)$, the roll inner-loop control system, and the transfer function from $\Delta\phi \rightarrow \Delta\psi$ for the outer-loop plant as shown in the following block diagram.



It should be noted that this heading hold assumes that the heading and yaw are the same, i.e. $\bar{\beta} = 0^\circ$ so $\sigma = \psi$, and that $\mu \approx \phi$ which is true for the linearized lateral-directional dynamics. Furthermore, the transfer function from $\Delta\phi \rightarrow \Delta\psi$ can be calculated using the steady flight relationship for the bank angle and the rate of change of heading/yaw, i.e.

$$\dot{\sigma} = \frac{g}{v_\infty} \tan \mu \quad (11.38)$$

which, for the small angle approximation $\tan \mu = \Delta\mu$ and using a stability frame for linearization, i.e. $v_\infty = \bar{u}$, one has

$$\Delta\psi(s) = \frac{g}{s\bar{u}}\Delta\phi(s) \quad (11.39)$$

after angle substitutions. It should be noted that often the roll angle must be hard limited for particular airplane missions. Thus, often one includes a **limiter** for the commanded roll which limits the maximum and minimum values to some specified values. Such an inclusion in the feedback control system will introduce a nonlinearity in the system, but generally reduces the speed of response for fast maneuvers. Performances of such systems is typically assessed through the nonlinear simulations of the feedback control systems initially designed with linear models.

References

For more information, please refer to the following

- Nelson, R. C., “8.4 Displacement Autopilot,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 292-312
- Schmidt, D. K., “12.2 Inner and Outer Loops, and Frequency Separation,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 703-706
- Schmidt, D. K., “12.5 Response Holds,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 708-737
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.6 Autopilots,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 322-344

11.3 Introductory Airplane Attitude Control Systems

Airplane Attitude Plant Model

The design of an airplane’s inner-loop attitude control systems for a particular steady-flight condition typically use the LTI longitudinal state-space model approximation, i.e.

$$\begin{aligned} \begin{bmatrix} \Delta\dot{u} \\ \Delta\dot{\alpha} \\ \Delta\dot{q} \\ \Delta\dot{\theta} \end{bmatrix} &= A_{long} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} + B_{long} \begin{bmatrix} \Delta\delta_e \\ \Delta\delta_T \end{bmatrix} \\ \Delta \vec{y}_{long} &= C_{long} \begin{bmatrix} \Delta u \\ \Delta\alpha \\ \Delta q \\ \Delta\theta \end{bmatrix} \end{aligned} \quad (11.40)$$

and the LTI lateral-directional state-space model approximation, i.e.

$$\begin{aligned} \begin{bmatrix} \Delta\dot{\beta} \\ \Delta\dot{p} \\ \Delta\dot{r} \\ \Delta\dot{\phi} \end{bmatrix} &= A_{lat} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} + B_{lat} \begin{bmatrix} \Delta\delta_a \\ \Delta\delta_r \end{bmatrix} \\ \Delta\vec{y}_{lat} &= C_{lat} \begin{bmatrix} \Delta\beta \\ \Delta p \\ \Delta r \\ \Delta\phi \end{bmatrix} \end{aligned} \quad (11.41)$$

where C_{long} and C_{lat} can be chosen for any state of interest.

Thus, the transfer function matrix for the longitudinal and lateral-directional inputs and outputs can be computed, respectively, as

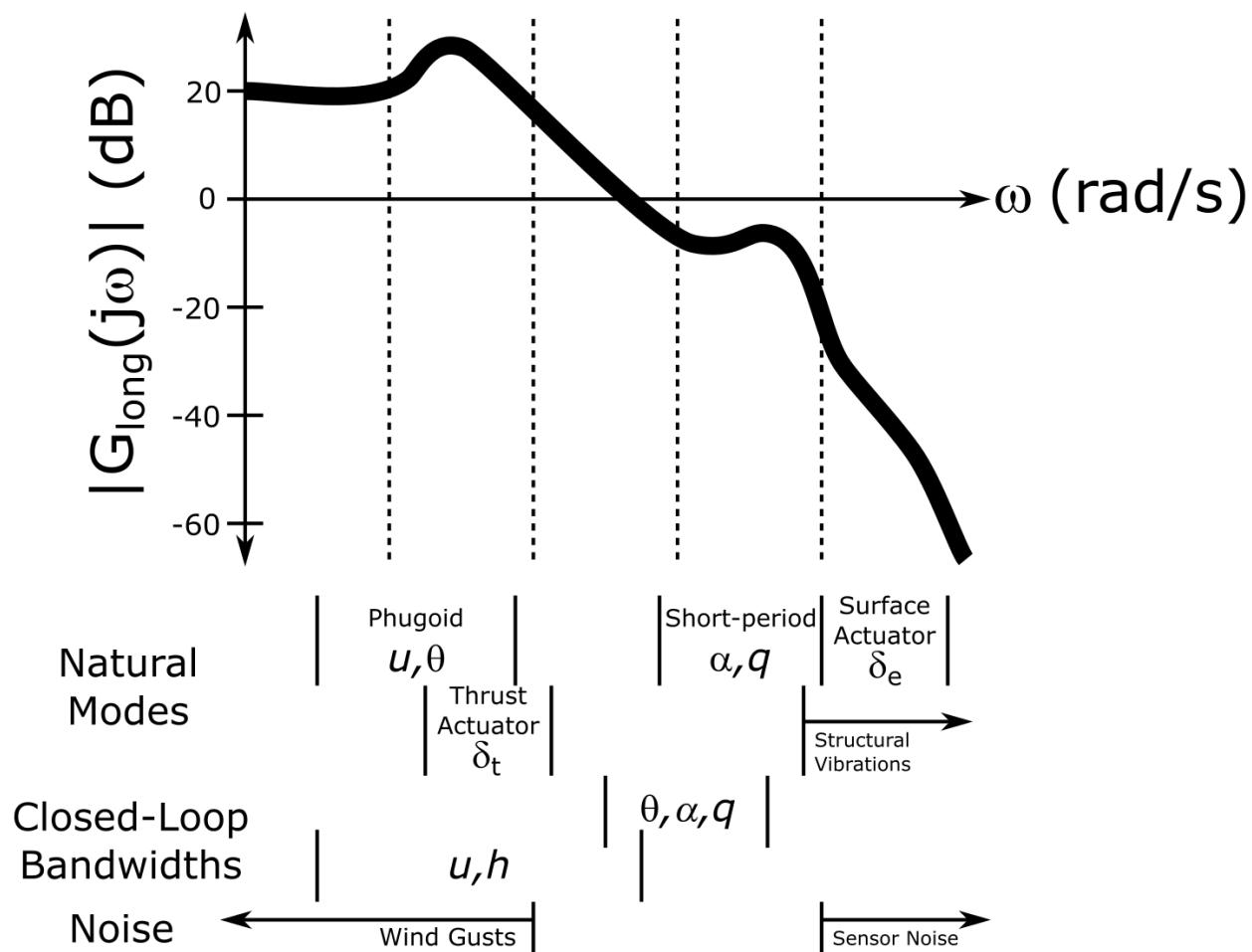
$$\Delta\vec{y}_{long}(s) = C_{long}(sI_{4\times 4} - A_{long})^{-1}B_{long} \begin{bmatrix} \Delta\delta_e(s) \\ \Delta\delta_T(s) \end{bmatrix} \quad (11.42)$$

and

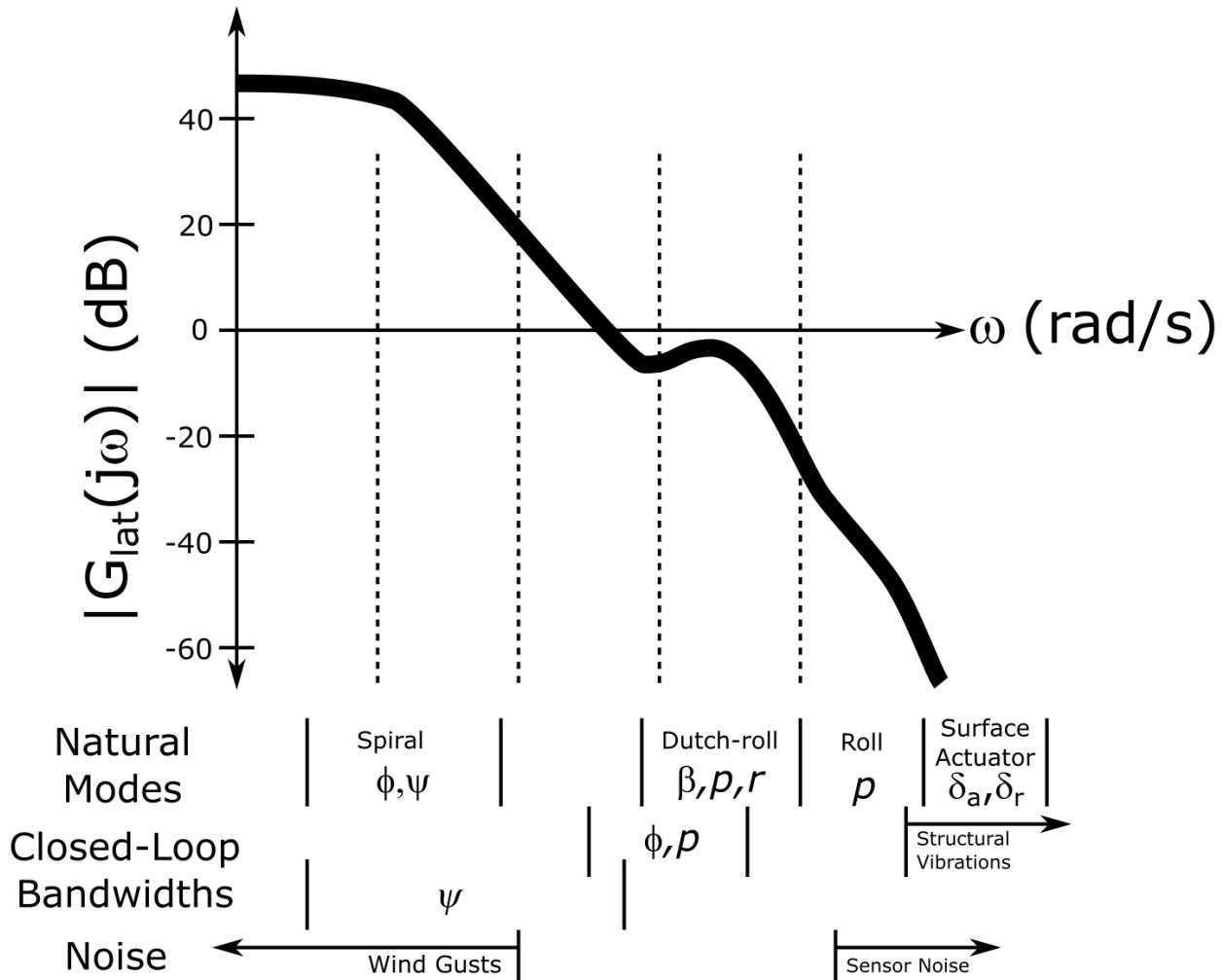
$$\Delta\vec{y}_{lat}(s) = C_{lat}(sI_{4\times 4} - A_{lat})^{-1}B_{lat} \begin{bmatrix} \Delta\delta_a(s) \\ \Delta\delta_r(s) \end{bmatrix} \quad (11.43)$$

where $(sI_{4\times 4} - A_{long})^{-1}$ provides the characteristic polynomial for the short-period and phugoid modes and $(sI_{4\times 4} - A_{lat})^{-1}$ provides the characteristic polynomial for the roll, dutch-roll, and spiral modes.

The general frequency response for these dynamic modes, the actuation modes, the wind disturbances, structural vibrations, the modes of the feedback control systems, and the frequency bands of the sensor noise are shown generally below for the longitudinal



and lateral-directional



It should be noted that the numerators of transfer functions to the states $\Delta u(s)$, $\Delta \alpha(s)$, $\Delta q(s)$, $\Delta \beta(s)$, $\Delta p(s)$, and $\Delta r(s)$ are third-order polynomials in s while $\Delta \theta(s)$ and $\Delta \phi(s)$ are second order polynomials in s as $\Delta q(s) = s\Delta \theta(s)$ and $\Delta p(s) = s\Delta \phi(s)$ for straight-and-level flight. Furthermore, though the linearized and decoupled transfer functions are used for airplane feedback control system design, it should be mentioned that the control inputs will generally affect *all* airplane states and should be simulated with the nonlinear airplane model once the guidance and control system design has been completed.

In addition, one may also use the following transfer function relationships via the small angle approximations in feedback control design. For the flight path angle, one has

$$\Delta \gamma(s) = \Delta \theta(s) - \Delta \alpha(s) \quad (11.44)$$

For altitude h , one has

$$\Delta h(s) = \frac{\bar{u}}{s} \Delta \gamma(s) \quad (11.45)$$

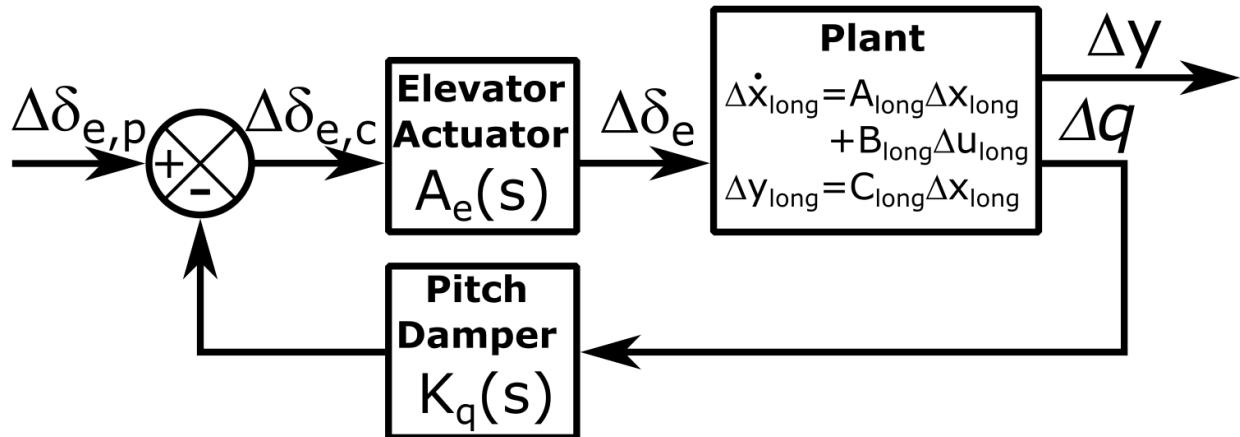
For the yaw angle, one has

$$\Delta\psi(s) = \frac{1}{s}\Delta r(s) \quad (11.46)$$

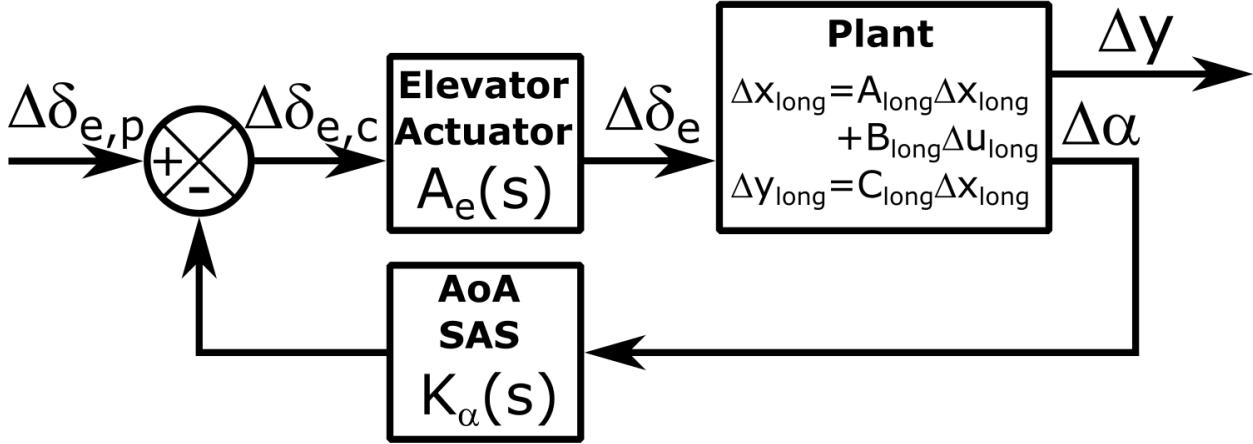
Airplane Stability Augmentation System Design

Stability augmentation systems (SAS) for airplanes are designed for the airplane to be controlled by a pilot (subscript p on the command), but the inherent modal stability or damping ratios are not within suitable flying qualities for the pilot. Thus, a SAS is used to augment the airplane dynamics to achieve certain stability or damping ratios for the airplane modes. These SAS are generally single-output feedback systems that use a single gain term, K_s , which changes the location of the system poles. To choose the value of K_s , one can use a **root locus** plot, i.e. a plot of the roots of the closed-loop characteristic polynomial in the complex plane as a function of K_s .

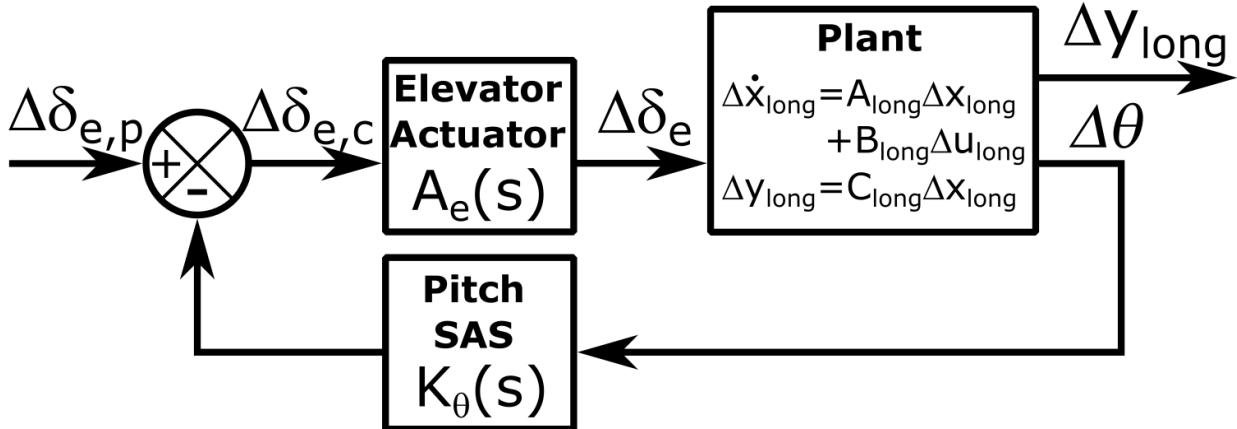
A **pitch damper** is used to reduce the damping ratio of the short-period mode of an airplane. A pitch damper, $K_q(s)$ uses a proportional gain, K_q , on the pitch rate, q , subtracted from the elevator deflection, δ_e . This SAS can be designed using a linearized plant as shown



An **angle of attack (AoA) SAS** is used to increase the frequency and stabilize the short-period mode of an airplane. An angle of attack SAS, $K_\alpha(s)$ uses a proportional gain, K_α , on the angle of attack, α , subtracted from the elevator deflection, δ_e . This SAS can be designed using a linearized plant as shown



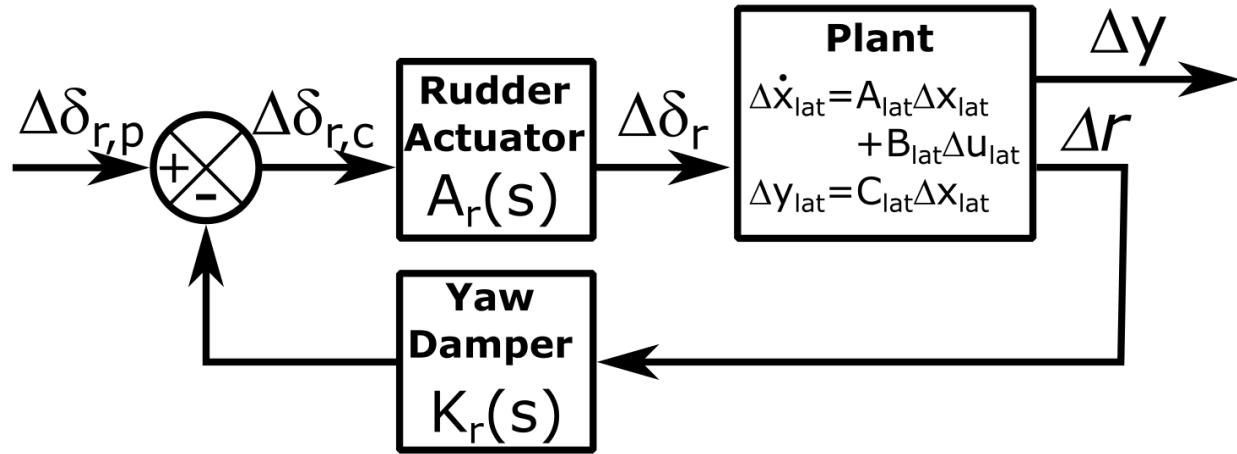
A **pitch SAS** is used to increase the natural frequency and stabilize the phugoid mode of an airplane. A pitch SAS, $K_\theta(s)$ uses a proportional gain, K_θ , on the pitch, θ , subtracted from the elevator deflection, δ_e . This SAS can be designed using a linearized plant as shown



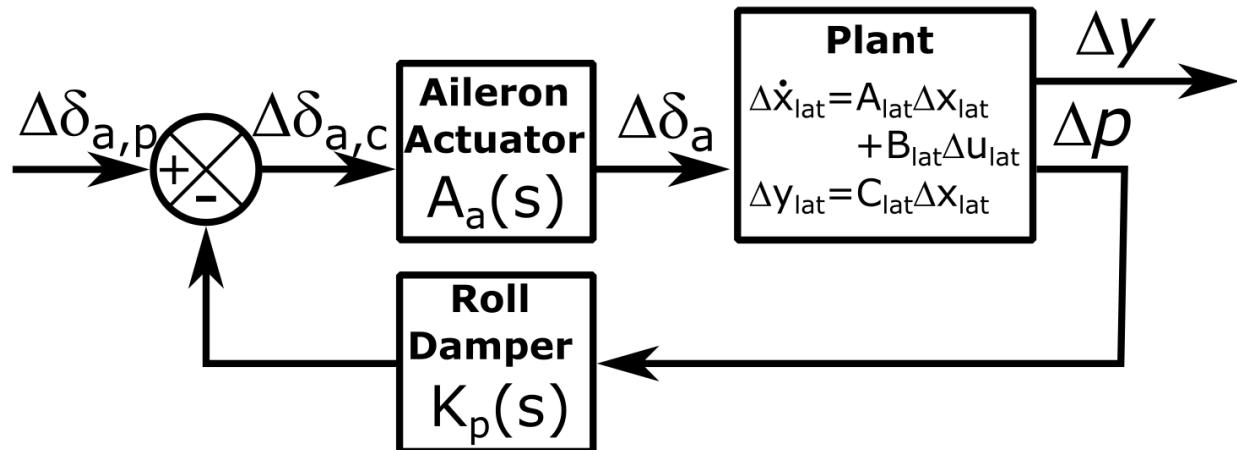
A **yaw damper** is used to increasing the damping ratio of the dutch-roll mode of an airplane. A first-order yaw damper feeds back the yaw rate, r to the system

$$K_r(s) = \frac{K_r s}{s + \omega_w} \quad (11.47)$$

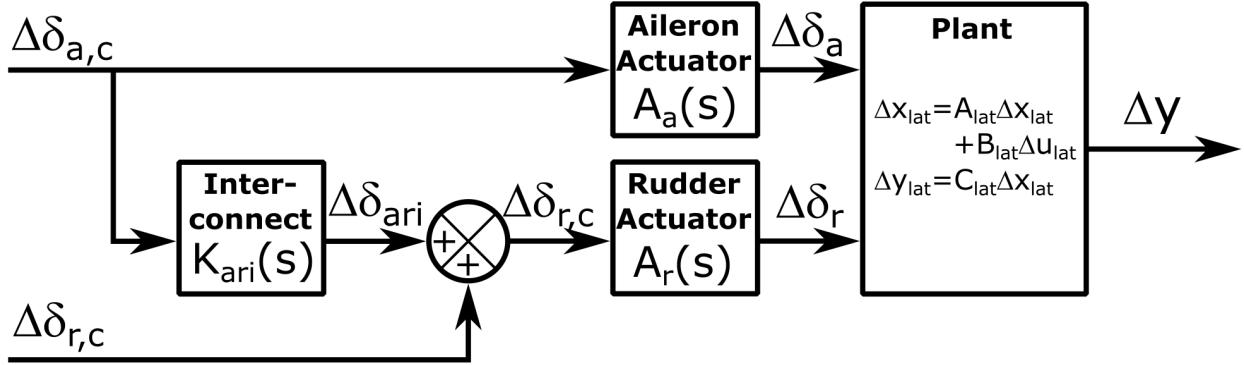
where K_r is the yaw damper gain and ω_w is the washout frequency which turns off the yaw damper for sustained turns where the steady yaw rate is not zero and is typically selected well below the dutch-roll natural frequency. The effect of this first-order filter “washes out” the yaw damper for low frequency signals, i.e. the yaw damper only affects $\omega \geq \omega_w$. The control input is subtracted from the rudder deflection, δ_r . This SAS can be designed using a linearized plant as shown



A **roll damper** is used to reduce the roll mode time constant which primarily affects Δp . A roll damper, $K_p(s)$ uses a proportional gain, K_p , on the roll rate, p , subtracted from the aileron deflection, δ_a . Thus, a roll damper can also be used to increase the effectiveness of a yaw damper as it feeds back Δp to the SAS. This SAS can be designed using a linearized plant as shown



As opposed to the longitudinal control inputs, the lateral-directional attitude control is performed by two control surface deflections which have strong coupling effects on the dynamics. For example, aileron deflections often cause an undesirable excitation of the dutch roll mode and/or an adverse yawing moment, N_{δ_a} . Thus, many aircraft typically use some sort of **aileron-rudder interconnect (ARI)** to reduce these effects.



A simple way to select K_{ari} is to cancel the adverse yaw with the rudder, i.e. forcing the effective $N_{\delta_a} \approx 0$ by setting the yawing moment due to rudder deflection equal to the negative of the yawing moment due to aileron deflection or in mathematical terms

$$N_{\delta_r}^* \Delta\delta_r = -N_{\delta_a}^* \Delta\delta_a \quad (11.48)$$

or letting

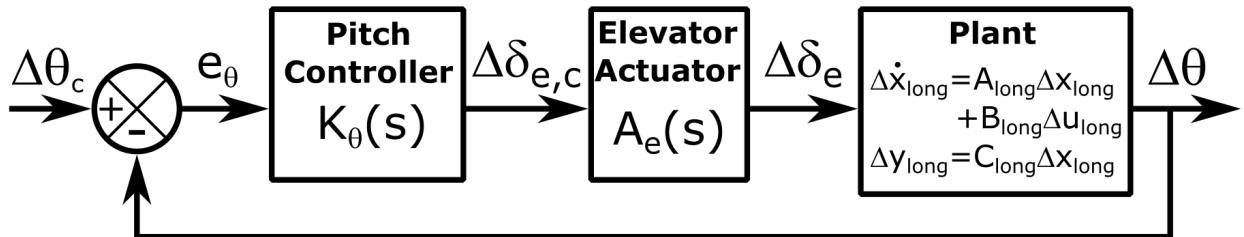
$$\Delta\delta_r = -\frac{N_{\delta_a}^*}{N_{\delta_r}^*} \Delta\delta_a \quad (11.49)$$

one has

$$K_{ari} = -\frac{N_{\delta_a}^*}{N_{\delta_r}^*} \quad (11.50)$$

Airplane Attitude Control System Design

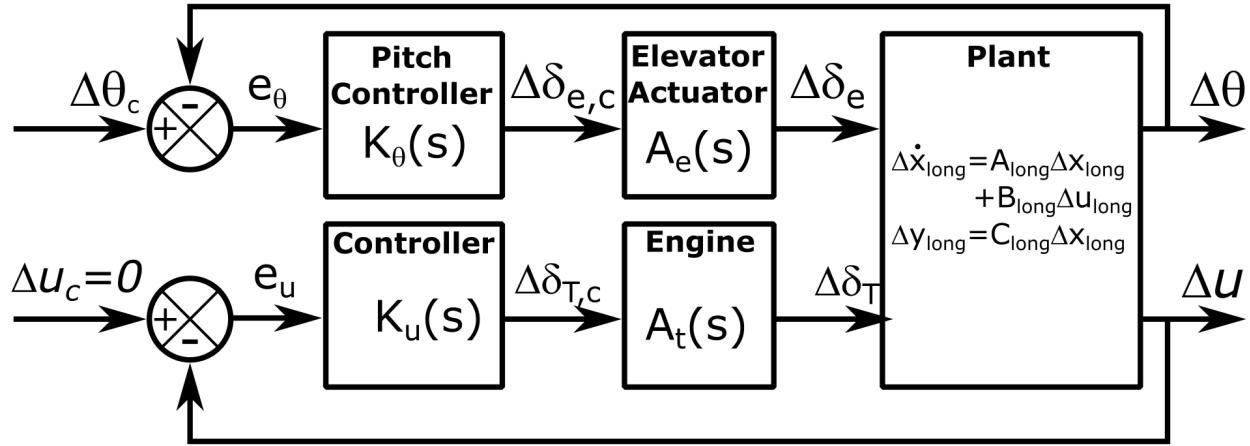
The design of the longitudinal inner-loop control system for an airplane typically uses the pitch control law, $K_\theta(s)$, with the longitudinal LTI plant model as shown in the following block diagram.



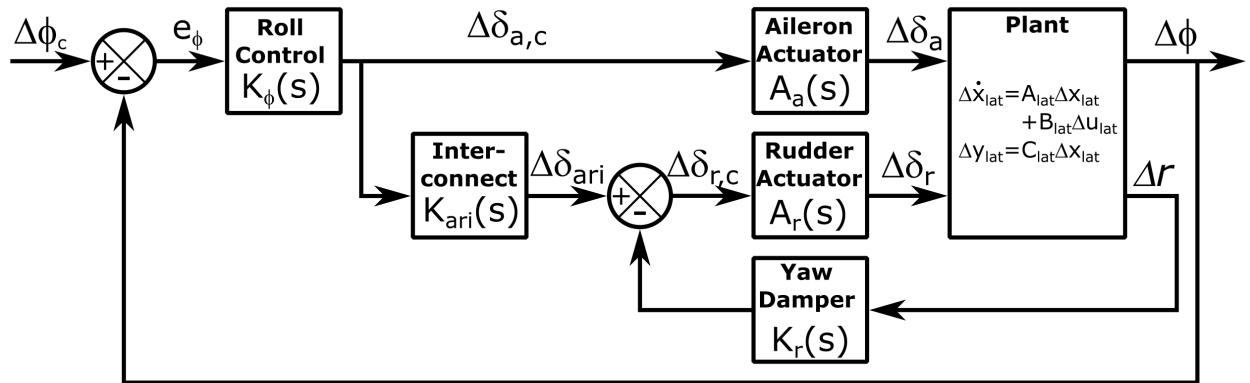
Other alternatives for the inner-loop include the pitch rate Δq and the angle of attack $\Delta\alpha$ as the single output for the system to track with the inner-loop. Note that a pitch damper can also be the “derivative stage” for a PID controller on θ .

When designing the pitch attitude controller, an important aspect to assess is whether an increase in $\Delta\theta_c$ leads to an increase or decrease in the flight-path angle $\Delta\gamma$ as $t \rightarrow \infty$. If the airplane is on the “front side

of the power curve,” i.e. a reduction in airspeed, v_∞ , requires less power to maintain level flight, then an increase in $\Delta\theta_c$ will lead to a positive steady-state $\Delta\gamma$. However, if the airplane is on the “back side of the power curve,” i.e. a reduction in airspeed, Δu , requires more power to maintain level flight, then an increase in $\Delta\theta_c$ will lead to a negative steady-state $\Delta\gamma$. To overcome this coupling between $\Delta\theta_c$, $\Delta\gamma$, and v_∞ , one can use δ_T in an outer-loop **auto-throttle** control system to keep the velocity at some Δu_c to account for the change in overall airspeed, v_∞ , due to a command in $\Delta\theta_c$ as shown



The lateral-directional inner-loop control system for an airplane typically uses the roll control law, $K_\phi(s)$, the yaw damper, $K_r(s)$, the ARI, $K_{ARI}(s)$, and the lateral-directional LTI plant model as shown in the following block diagram.



This type of roll controller can be used to command a coordinated turn, i.e. $\beta = 0^\circ$ with $\Delta\phi_c \neq 0^\circ$ through the use of the ARI. It should be noted that the washout filter stage must be included for the case of sustained turns where the yaw rate is not zero. Without it, the yaw rate feedback (i.e. damper) would tend to “fight” the turn by maintaining a rudder deflection opposite to that desired for the turn. Considering these additions to a roll angle feedback control system, one typically must first form the plant model $G_\phi(s)$ using

the yaw damper before loop-shaping $L_\phi(s) = G_\phi(s)K_\phi(s)$. It should be mentioned that though using an ARI is one method for controlling a coordinated turn, i.e. $\beta = 0^\circ$ with $\Delta\phi_c \neq 0^\circ$, one may also consider feeding back to the rudder, either β , a_y , or r .

Lastly, when performing a coordinated turn, one must also generate the suitable longitudinal control inputs to maintain altitude, a concept called **turn compensation**. Primarily, by rotating the lift vector of the airplane, the vertical component of lift must still counteract the weight for constant altitude. Thus, as an airplane enters the turn, the angle of attack must be increased which requires an appropriate elevator deflection given the flight velocity and bank angle. One simple method is to compute the yaw rate and roll angle to generate the necessary commanded pitch rate for a pitch or pitch rate feedback control system, i.e.

$$q_c = r \tan \phi \quad (11.51)$$

where these values are for the entire airplane, not the perturbed states.

References

For more information, please refer to the following

- Nelson, R. C., “8.2 Aircraft Transfer Functions,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 283-288
- Nelson, R. C., “8.4 Displacement Autopilot,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 292-312
- Nelson, R. C., “8.5 Stability Augmentation,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 312-314
- Schmidt, D. K., “12.3 The Flight-Dynamics Frequency Spectra,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 706-708
- Schmidt, D. K., “12.4 Attitude Control,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 708-737
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.4 Stability Augmentation,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 287-303
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.5 The Handling-Qualities Requirements,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 303-322
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.6 Autopilots,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 322-344

11.4 Spacecraft Dual-Spin Stabilization Systems

In order to control the attitude of a satellite, one must change the dynamics of the angular velocity in some way. As the angular velocity is directly related to the angular momentum, one requires momentum exchange devices to change the angular momentum without a change in mass to affect the attitude dynamics. The simplest stabilization approach uses a single secondary momentum wheel to stabilize an axisymmetric, spinning spacecraft or a gravity-gradient-stabilized satellite which is known as **dual-spin stabilization**.

Gyrostat Stabilization Systems

A **gyrostat** configuration uses an external prolate momentum device, called the rotor, joined to a *smaller* axisymmetric platform along a colinear spin axis at a bearing. As a rigid body, the platform and rotor have the same perpendicular angular velocity, $\vec{\omega}_\perp$, but have different components of angular velocity, along the spin axis \hat{k}_B , where the platform's spin rate is designed much slower than the rotor's spin rate. Also, notably the bearing must have an electric motor integrated in order to overcome the frictional resistance which would eventually cause the platform to match the rotor.

Under torque-free motion, the gyrostat attitude dynamics, taking the axisymmetric platform as the “wheel” with angular velocity, ω_1 along the spin axis and motor-driven axisymmetric rotor as the body-fixed frame for the vehicle with angular velocity $\vec{\omega}$, one has

$$\dot{\vec{H}}_G^{(v)} + [\vec{\omega}]_\times \vec{H}_{G,v} + \dot{\vec{H}}_{G_1}^1 + [\vec{\omega}]_\times \vec{H}_{G_1}^1 = 0 \quad (11.52)$$

The total angular momentum about G is

$$\vec{H}_{G,v} = (I_G + I_{G,m_1}) \vec{\omega} \quad (11.53)$$

where I_G is the inertia tensor of the rotor about the center of mass of the platform-rotor system, G , and I_{G,m_1} is the inertia tensor of the concentrated mass of the platform about G . The angular momentum of the platform about its own center of mass is

$$\vec{H}_{G_1} = I_{G_1} \vec{\omega}_1 \quad (11.54)$$

The relative derivatives of the angular momentum are given

$$\dot{\vec{H}}_{G,v} = (I_G + I_{G,m_1}) \dot{\vec{\omega}} \quad (11.55)$$

and

$$\dot{\vec{H}}_{G_1} = I_{G_1} \dot{\vec{\omega}}_1 \quad (11.56)$$

as both the platform and rotor are axisymmetric. Thus, the gyrostat attitude dynamics can be written as

$$(I_{G,v} + I_{G,m_1}) \dot{\vec{\omega}} + [\vec{\omega}]_\times (I_{G,v} + I_{G,m_1}) \vec{\omega} + I_{G_1} \dot{\vec{\omega}}_1 + [\vec{\omega}]_\times I_{G_1} \vec{\omega}_1 \quad (11.57)$$

Then, defining the relative angular velocity of the platform to the rotor as ω_{rel} , and the common $\vec{\omega}_\perp$, one

$$\vec{\omega}_{rel} = \vec{\omega}_1 - \vec{\omega} = \begin{bmatrix} p \\ q \\ r + \omega_{rel} \end{bmatrix} - \begin{bmatrix} p \\ q \\ r \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \omega_{rel} \end{bmatrix} \quad (11.58)$$

one has

$$(I_{G,v} + I_{G,1}) \vec{\omega} + [\vec{\omega}] \times (I_{G,v} + I_{G,1}) \vec{\omega} + I_{G_1} \vec{\omega}_{rel} + [\vec{\omega}] \times I_{G_1} \vec{\omega}_{rel} \quad (11.59)$$

where $I_{G,1} = I_{G,m_1} + I_{G_1}$ by the parallel axis formula.

This expression can be expanded by defining the components of the inertia tensor for the rotor about the system center of mass as

$$I_{G,v} = \begin{bmatrix} I_{1,v} & 0 & 0 \\ 0 & I_{1,v} & 0 \\ 0 & 0 & I_{3,v} \end{bmatrix} \quad (11.60)$$

the components of the inertia tensor for the platform about the system center of mass as

$$I_{G,1} = \begin{bmatrix} I_{1,1} & 0 & 0 \\ 0 & I_{1,1} & 0 \\ 0 & 0 & I_{3,1} \end{bmatrix} \quad (11.61)$$

and the components of the inertia tensor for the platform about its own center of mass as

$$I_{G_1} = \begin{bmatrix} I_{1,1}^* & 0 & 0 \\ 0 & I_{1,1}^* & 0 \\ 0 & 0 & I_{3,1}^* \end{bmatrix} \quad (11.62)$$

where $I_{3,1} = I_{3,1}^*$ as G and G_p lie on the same z_B -axis.

With these definitions and the definitions, one has

$$\begin{bmatrix} (I_{1,v} + I_{1,1}) \dot{p} \\ (I_{1,v} + I_{1,1}) \dot{q} \\ (I_{1,v} + I_{3,1}) \dot{r} \end{bmatrix} + \begin{bmatrix} (I_{3,v} - I_{1,v} + I_{3,1} - I_{1,1}) qr \\ (I_{1,v} - I_{3,v} + I_{1,1} - I_{3,1}) pr \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ I_{3,1} \dot{\omega}_{rel} \end{bmatrix} + \begin{bmatrix} I_{3,1} q \omega_{rel} \\ -I_{3,1} p \omega_{rel} \\ 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (11.63)$$

or

$$\begin{bmatrix} I_1 \dot{p} + (I_3 - I_1) qr \\ I_1 \dot{q} + (I_1 - I_3) pr \\ I_3 \dot{r} \end{bmatrix} + \begin{bmatrix} I_{3,1} q \omega_{rel} \\ -I_{3,1} p \omega_{rel} \\ I_{3,1} \dot{\omega}_{rel} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (11.64)$$

where $I_1 = I_{1,v} + I_{1,1}$ and $I_3 = I_{3,v} + I_{3,1}$. These are three differential equations in four states, p , q , r , and ω_{rel} . As mentioned, for dual-spin stabilization the electric motor at the bearing is used to set ω_{rel} , thus $\omega_{rel} = 0$. Thus, $r = \bar{r}$ and one can change p and q as function of the relative spin rate ω_{rel} .

Similar to the single-spinner, one can analyze this dual-spinner using an energy dissipation model. For the entire gyrostat, the angular momentum is

$$\vec{H}_G = \begin{bmatrix} I_{1,v} p \\ I_{1,v} q \\ I_{3,v} r \end{bmatrix} + \begin{bmatrix} I_{1,1} p \\ I_{1,1} q \\ I_{3,1}(r + \omega_{rel}) \end{bmatrix} \quad (11.65)$$

or with $r_1 = r + \omega_{rel}$, one has

$$\vec{H}_G = \begin{bmatrix} I_1 p \\ I_1 q \\ I_{3,v} r + I_{3,1} r_1 \end{bmatrix} \quad (11.66)$$

with magnitude

$$\|\vec{H}_G\|_2^2 = I_1 \omega_{\perp}^2 + (I_{3,v}r + I_{3,1}r_1)^2 \quad (11.67)$$

where $\omega_{\perp}^2 = p^2 + q^2$. For torque-free motion, one has

$$\frac{d\|\vec{H}_G\|_2^2}{dt} = 0 = I_1 \frac{d\omega_{\perp}^2}{dt} + 2(I_{3,v}r + I_{3,1}r_1)(I_{3,v}\dot{r} + I_{3,1}\dot{r}_1) \quad (11.68)$$

or

$$\frac{d\omega_{\perp}^2}{dt} = -\frac{2}{I_1^2} (I_{3,v}r + I_{3,1}r_1)(I_{3,v}\dot{r} + I_{3,1}\dot{r}_1) \quad (11.69)$$

In addition, the total rotational kinetic energy of the gyrostat

$$T_R = T_{R,0} + T_{R,1} = \frac{1}{2}I_1\omega_{\perp}^2 + \frac{1}{2}I_{3,v}r^2 + \frac{1}{2}I_{3,1}r_1^2 \quad (11.70)$$

where $T_{R,0}$ is the energy of the rotor and $T_{R,1}$ is the energy of the platform. Differentiating this with respect to time and rearranging, one has

$$\frac{d\omega_{\perp}^2}{dt} = \frac{2}{I_1^2} (\dot{T}_{R,0} + \dot{T}_{R,1} - I_{3,v}r\dot{r} - I_{3,1}r_1\dot{r}_1) \quad (11.71)$$

Equating this with the expression from the angular momentum, one has

$$\frac{2}{I_1^2} (\dot{T}_{R,0} + \dot{T}_{R,1} - I_{3,v}r\dot{r} - I_{3,1}r_1\dot{r}_1) = -\frac{2}{I_1^2} (I_{3,v}r + I_{3,1}r_1)(I_{3,v}\dot{r} + I_{3,1}\dot{r}_1) \quad (11.72)$$

Rearranging, one has

$$\dot{T}_{R,0} + \dot{T}_{R,1} = \frac{I_{3,v}}{I_1} ((I_1 - I_{3,v})r - I_{3,1}r_1)\dot{r} + \frac{I_{3,1}}{I_1} ((I_1 - I_{3,1})r_1 - I_{3,v}r)\dot{r}_1 \quad (11.73)$$

Equating the two terms to the energy dissipation, one has

$$\dot{r} = \frac{I_1}{I_{3,v}(I_1 - I_{3,v})r - I_{3,1}r_1} \frac{\dot{T}_{R,0}}{\dot{T}_{R,1}} \quad (11.74)$$

and

$$\dot{r}_1 = \frac{I_1}{I_{3,1}(I_1 - I_{3,1})r_1 - I_{3,v}r} \frac{\dot{T}_{R,1}}{\dot{T}_{R,0}} \quad (11.75)$$

which allows one to rewrite

$$\frac{d\omega_{\perp}^2}{dt} = \frac{2}{I_1} \left(\frac{\dot{T}_{R,0}}{I_{3,1}\frac{r_1}{r} - (I_1 - I_{3,v})r} + \frac{\dot{T}_{R,1}}{I_{3,v} - (I_1 - I_{3,1})\frac{r_1}{r}} \right) \left(I_{3,v} + I_{3,1}\frac{r_1}{r} \right) \quad (11.76)$$

An important class of dual-spin stabilized satellites exists where $\frac{r_1}{r} \approx 0$, e.g. a geosynchronous satellite

$$\frac{r_1}{r} = \frac{2\pi rad/day}{2\pi rad/s} \approx 10^{-5} \quad (11.77)$$

and for interplanetary spacecraft, $r_1 = 0$. Here, one has the simpler

$$\frac{d\omega_{\perp}^2}{dt} = \frac{2}{I_1} \left(\dot{T}_{R,0} + \frac{I_{3,v}}{I_{3,v} - I_1} \dot{T}_{R,1} \right) \quad (11.78)$$

Thus, as $\dot{T}_{R,0} < 0$ and $\dot{T}_{R,1} < 0$, if the rotor is oblate, i.e. $I_{3,v} > I_1$, then the satellite is unconditionally stable. However, if the rotor is prolate, i.e. $I_{3,v} < I_1$, then the satellite is stable, if and only if

$$\dot{T}_{R,1} < \frac{I_{3,v}}{I_{3,v} - I_1} \dot{T}_{R,0} \quad (11.79)$$

where the platform dissipation rate is typically augmented by passive nutation dampers.

Momentum-Biased Stabilization

Momentum-biased satellites are dual-spin stabilized satellites that do not consist of two external parts, but use an internal momentum wheel as a momentum exchange device to stabilize the satellite in the presence of moment disturbances.

References

For more information, please refer to the following

- Curtis, H. D., “10.4 Dual-Spin Spacecraft,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 476-486
- Sidi, M. J., “6.7 Dual-Spin Stabilization,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 148-151
- Sidi, M. J., “8.3 Stabilization with Active Control,” *Spacecraft Dynamics and Control: A Practical Engineering Approach*, Cambridge University Press, Cambridge, 2000, pp. 214-222

11.5 Precision Approach Guidance Systems

Guided Precision Approaches

In commercial aviation, guidance systems have been developed to assist an aircraft during the approach and landing phases of an aircraft’s flight as these phases require the most accurate navigation information and precision guidance especially when visibility is impaired for pilots and an autopilot is engaged, also known as a **precision approach**. These systems are designed to supply this information using radio signals that are interpreted by specialized equipment onboard the aircraft which supply a heading correction that the pilot or autopilot must make to continue on the prescribed descent trajectory to that airport’s runway. Thus, these systems are maintained by individual airports. Thus, the reference trajectories generated by these “guidance” systems are constant, predefined trajectories based on the topography of the area, the layout of the airport and runways, and regulations concerning safety for aircraft descent.

An important parameter in precision approaches is the **decision height (DH)** or **decision altitude (DA)**. Both of which are defined as the specified lowest height in the approach descent at which, if the required

visual reference to continue the approach is not visible to the pilot, the pilot must initiate a missed approach maneuver and reroute to try the approach again. The visual reference is typically the runway markings or runway environment. A decision height is measured *above ground level (AGL)* while a decision altitude is measured above *mean sea level (MSL)*. The specific values for DH at a given airport are established with intention to allow a pilot sufficient time to safely re-configure an aircraft to climb and execute the missed approach procedures while avoiding terrain and obstacles. A DH denotes the height in which a missed approach procedure must be started, it does not preclude the aircraft from descending below the prescribed DH.

For autonomous guided approaches, there are four categories which allow these decision heights to be lowered.

- For Category I, the decision height is not lower than 200 feet and the runway visual range is not less than 550 meters.
- For Category II, the decision height is below 200 ft but not lower than 100 feet and the runway visual range is not less than 300 meters.
- For Category III A, the decision height is lower than 100 feet, and the runway visual range is not less than 200 meters.
- For Category III B, the decision height is lower than 50 ft, or no decision height, and the runway visual range is lower than 200 meters but not less than 75 meters.

To allow these guided precision approaches, the following technologies serve commercial aircraft systems as both navigational and guidance aids.

One of the earliest technologies for automatic guidance and is still in use today is the **Instrument Landing System (ILS)** which used radionavigation to provide aircraft with horizontal and vertical guidance information during approach and landing. At certain fixed points, it also provides the distance to the reference point of landing. An ILS uses two signals: a **localizer** and a **glide-slope** which indicate the correction that the pilot or autopilot must make. As part of this system, there are also typically **marker beacons** which provide distance to the runway information at setpoints along the approach. These setpoints are published on the airport documentation for the approach.

This is displayed to pilots as a point plotted on 2 coordinate axes whose origin is “0” error. If the aircraft is to the left of the reference trajectory, the point is to the right of the origin. If the aircraft is above the set glide-slope path, then the point is below the origin. The glide-slope path for an ILS is typically defined as 3° above the horizontal.

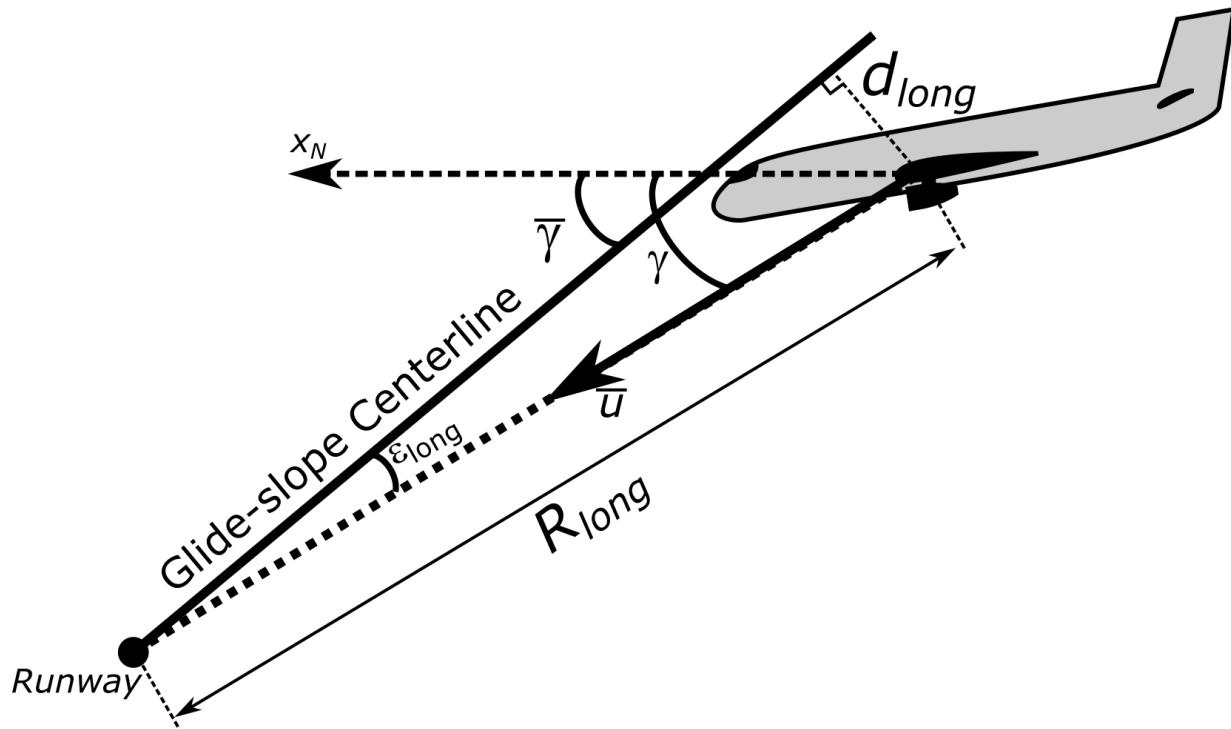
Occasionally a modified ILS called an **Instrument Guidance System (IGS)**, also known as a **Localizer-type Directional Aid (LDA)** in the United States, must be used for non-straight approaches into airports with certain topographical features which prevent normal operation. One of the most famous of these was runway 13 at Kai Tak International Airport in Hong Kong due to the mountains, apartments, and ocean surrounding the airport. **Distance measuring equipment (DME)** is also a common system available at airports that provides pilots or autopilots with a distance to runway measurement in nautical miles which can be deployed with or without an ILS and are rapidly replacing marker beacons.

Due to the rapid development of Global Navigation Satellite Systems (GNSS), in particular the Global Positioning System (GPS) run by the United States Department of Defense, the most recent advances in

automatic guidance technologies for aircraft include the following technologies which enhance the basic capabilities of GNSS by reducing the errors found in the GNSS signals thereby improving the accuracy of the navigation solution. The need and details for these technologies will be discussed in later parts of this textbook. Localizer Performance with Vertical guidance (LPV) system is based on the Satellite-Based Augmentation System (SBAS) to GNSS measurements, which for GPS is also known as the Wide Area Augmentation System (WAAS). This provides greater accuracy than GPS alone which is necessary for guidance during aircraft approaches, but it is not as accurate as the next system. The Ground-Based Augmentation System (GBAS), which for GPS is also known as Local Area Augmentation System (LAAS), also augments the GNSS measurements in order to provide enhanced levels of service to support automatic guidance information during all phases of approach, landing, departure, and surface operations within radio distance. GBAS is anticipated to be able to achieve Category III B guided precision approaches and is expected to replace ILS in the future.

Glide-Slope Guidance System

For precision approach guidance, an airplane is to follow a “straight-line” path in the longitudinal plane for the glide-slope



where d_{long} is the longitudinal position deviation, ϵ_{long} is the longitudinal angular deviation, R_{long} is the longitudinal range to intercept, $\bar{\gamma}$ is the reference glide-slope angle.

Given this figure, the longitudinal deviations can be related by

$$\sin \epsilon_{long} = \frac{d_{long}}{R_{long}} \quad (11.80)$$

which for small angular deviations is

$$\epsilon_{long} = \frac{d_{long}}{R_{long}} \quad (11.81)$$

Furthermore, as the longitudinal angular deviation is related to the flight path angle by

$$\epsilon_{long} = \bar{\gamma} - \gamma = -\Delta\gamma \quad (11.82)$$

Then, by rearrangement and substitution, one has

$$d_{long} = -R_{long}\Delta\gamma \quad (11.83)$$

then taking the derivative, one has

$$\dot{d}_{long} = -\dot{R}_{long}\Delta\gamma \quad (11.84)$$

and the instantaneous range rate is

$$\dot{R}_{long} = -\bar{u} \quad (11.85)$$

one has for the measured angular deviation

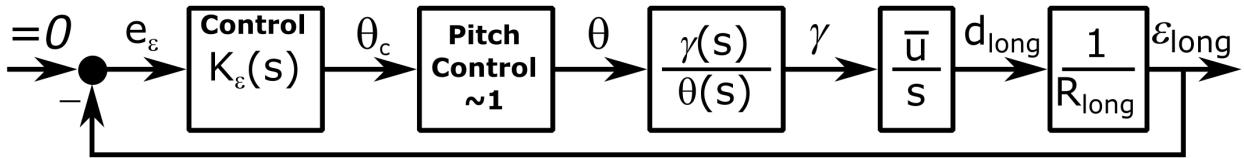
$$\dot{\epsilon}_{long} \approx \frac{\bar{u}}{R_{lat}}\Delta\gamma \quad (11.86)$$

In the Laplace domain (dropping the Δ)

$$\frac{\epsilon_{long}(s)}{\Gamma(s)} \approx \frac{\bar{u}}{Rs} \quad (11.87)$$

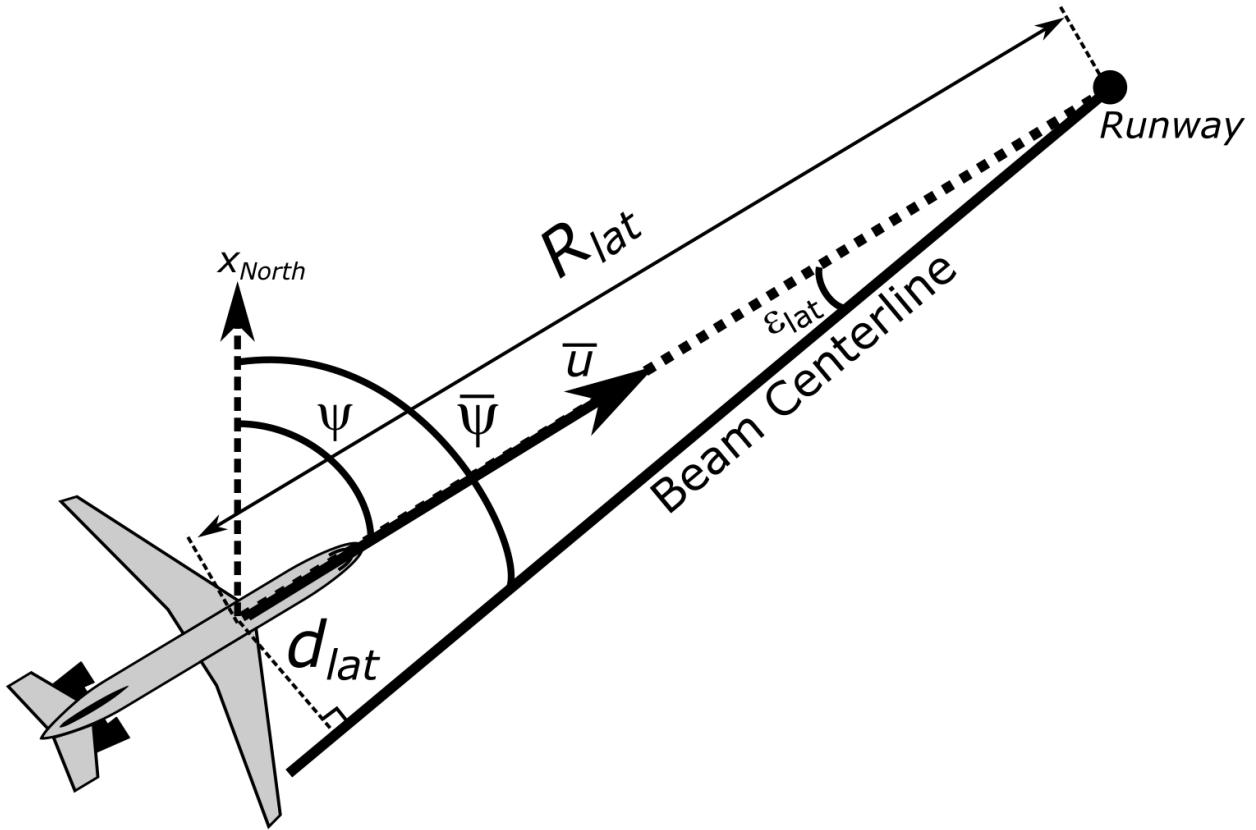
where it should be noted that this also assumes that the body-fixed reference frame is the stability frame, i.e. $\bar{\alpha} = 0$, thus $\bar{\gamma} = \bar{\theta}$.

Thus, one can form a block diagram of the **glide-slope guidance system** as



which is very similar to the altitude hold feedback loop with an additional outer-loop guidance for reducing the angular deviation from a reference flight-path to zero which scales inversely with the distance R_{long} from the runway.

Localizer Guidance



where d_{lat} is the lateral-directional position deviation, ϵ_{lat} is the lateral-directional angular deviation, R_{lat} is the lateral-directional range to intercept, $\bar{\psi}$ is the reference beam centerline for the localizer.

Given this figure, the lateral-directional deviations can be related by

$$\sin \epsilon_{lat} = \frac{d_{lat}}{R_{lat}} \quad (11.88)$$

which for small angular deviations is

$$\epsilon_{lat} = \frac{d_{lat}}{R_{lat}} \quad (11.89)$$

Furthermore, as the lateral angular deviation is related to the heading angle by

$$\epsilon_{lat} = \psi - \bar{\psi} \quad (11.90)$$

Then, by rearrangement and substitution, one has

$$d_{lat} = -R_{lat} (\psi - \bar{\psi}) \quad (11.91)$$

then taking the derivative, one has

$$\dot{d}_{lat} = -\dot{R}_{lat} (\psi - \bar{\psi}) \quad (11.92)$$

and the instantaneous range rate is

$$\dot{R}_{lat} = -\bar{u} \quad (11.93)$$

one has for the measured angular deviation

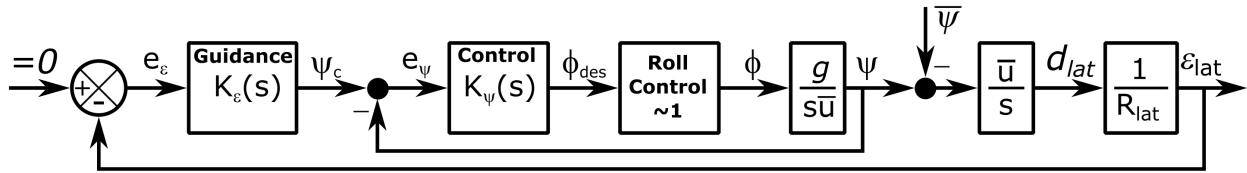
$$\dot{\epsilon}_{lat} \approx \frac{\bar{u}}{R_{lat}} (\psi - \bar{\psi}) \quad (11.94)$$

or in the Laplace domain

$$\frac{\epsilon_{lat}(s)}{\Psi(s) - \bar{\Psi}(s)} \approx \frac{\bar{u}}{Rs} \quad (11.95)$$

where it should be noted that this also assumes that the body-fixed reference frame is the stability frame, i.e. $\vec{v} = u$ and that coordinated flight is occurring, i.e. $\bar{\beta} = 0^\circ$.

Thus, one can form a block diagram of the **localizer guidance system** as



which essentially uses a heading hold feedback loop with an additional outer-loop guidance for reducing the angular deviation from a reference heading to zero which scales inversely with the distance R_{lat} from the runway.

References

For more information, please refer to the following

- Nelson, R. C., “8.6 Instrument Landing,” *Flight Stability and Automatic Control*, 2nd ed., Vol 1., McGraw-Hill, New York, 1997, pp. 314-318
- Schmidt, D. K., “12.6 Path Guidance - ILS Couplers and VOR Homing,” *Modern Flight Dynamics*, 1st ed., Vol. 1, McGraw-Hill, New York, 2012, pp. 753-771
- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “4.6 Autopilots,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 322-344

Advanced Aerospace Vehicle Control Systems

12.1 Aerospace Structural Mode Control Systems

12.2 Airplane Total Energy and Heading Control Systems

In classical airplane attitude control design, one uses SISO principles to design the pitch attitude controller using the elevator input, the roll attitude controller using the aileron control input, sideslip attitude controller using the rudder control input, and the airspeed controller using the thrust input. However, this decoupled approach is sub-optimal as these SISO systems do not coordinate their control actions to manage the airplane energy or heading for a commanded trajectory. In particular, the airspeed controller does not incorporate the important state variable, the flight-path angle, in determining the required thrust while the pitch attitude controller does not know the vertical velocity nor its limits. Furthermore, the roll and sideslip angles are interrelated to the yaw and heading angles, one is required to use a yaw damper and turn compensation for coordinated turns in the inner-loop as well as thrust asymmetry compensation in the event of an engine-out condition.

As the lift force for airplanes is affected by both speed and angle of attack, the fundamental physics in longitudinal control should be energy management of the flight vehicle. Also, one can use the small angle dynamics to provide processed error signals that decouple the yaw and sideslip before providing the command to the ailerons and rudder actuator using the fundamental physics of the lateral and directional angles of the flight vehicle. Thus, modern fixed-wing vehicles often use total energy and total heading control systems for MIMO attitude control.

Point-Mass Energy Dynamics

From the point-mass perspective with the constant-wind assumption in straight flight, the total energy of the flight vehicle, \mathcal{E} , is given by the sum of the kinetic and potential energy, i.e.

$$\mathcal{E} = mg \left(\frac{1}{2} \frac{v_\infty^2 + v_w^2}{g} + h \right) \quad (12.1)$$

where m is the mass, v_∞ is the airspeed, v_w is the constant wind, g is the assumed constant acceleration due to gravity, and h is the altitude. Similarly, the Lagrangian, \mathcal{L} , is given by the difference of the kinetic and potential energy, i.e.

$$\mathcal{L} = mg \left(\frac{1}{2} \frac{v_\infty^2 + v_w^2}{g} - h \right) \quad (12.2)$$

Differentiating with respect to time, one has

$$\dot{\mathcal{E}} = mg \left(\frac{v_\infty \dot{v}_\infty}{g} + \dot{h} \right) \quad (12.3)$$

and

$$\dot{\mathcal{L}} = mg \left(\frac{v_\infty \dot{v}_\infty}{g} - \dot{h} \right) \quad (12.4)$$

Rearranging and substituting for $\dot{h} \approx \bar{v}_\infty \gamma$ for the small angle approximation, one can rewrite these equations as

$$\dot{\tilde{\mathcal{E}}} = \frac{\dot{\mathcal{E}}}{mg v_\infty} = \frac{\dot{v}_\infty}{g} + \gamma \quad (12.5)$$

and

$$\dot{\tilde{\mathcal{L}}} = \frac{\Delta \dot{\mathcal{L}}}{mg v_\infty} = \frac{\dot{v}_\infty}{g} - \gamma \quad (12.6)$$

where $\dot{\tilde{\mathcal{E}}}$ is the normalized total energy and $\dot{\tilde{\mathcal{L}}}$ is the normalized Lagrangian.

Next, recall the x_W -axis force balance is given by

$$-D + T \cos \alpha \cos \beta - mg \sin \gamma = m \dot{v}_\infty \quad (12.7)$$

which for small angles, one has

$$T = mg \left(\frac{\dot{v}_\infty}{g} + \gamma \right) + D \quad (12.8)$$

For some steady-flight conditions, one has

$$\bar{T} + \Delta T = mg \left(\frac{\dot{v}_\infty}{g} + \bar{\gamma} + \Delta \gamma \right) + \bar{D} - \Delta D \quad (12.9)$$

where assuming $\bar{T} \approx mg \bar{\gamma} + \bar{D}$ for trim and $\Delta D \approx 0$, one has

$$\frac{\Delta T}{mg} = \delta_T = \frac{\dot{v}_\infty}{g} + \Delta \gamma = \frac{\dot{\tilde{\mathcal{E}}}}{mg v_\infty} \quad (12.10)$$

which demonstrates that the normalized thrust, δ_T , can be used effectively to change the rate of the total energy for the system.

Finally, recall that for coordinated, straight-flight, one has

$$\gamma = \theta - \alpha \quad (12.11)$$

Thus, the normalized Lagrangian can be written as

$$\dot{\tilde{\mathcal{L}}} = \frac{\dot{v}_\infty}{g} + \alpha - \theta \quad (12.12)$$

which demonstrates that the pitch, θ , can be used effectively to change the rate of the normalized Lagrangian. In TECS, the elevator deflection, δ_e , provides an additional SISO inner-loop attitude control for the pitch.

Total Energy Control System Design

In a total energy control system (TECS) design, the planning system provides the TECS a commanded airspeed acceleration in g's, $\frac{\dot{v}_{\infty,c}}{g}$, and a commanded flight-path angle, γ_c . This is related to the a commanded total energy rate

$$\dot{\tilde{\mathcal{E}}}_c = \frac{\dot{v}_{\infty,c}}{g} + \gamma_c \quad (12.13)$$

and a commanded Lagrangian rate

$$\dot{\tilde{\mathcal{L}}}_c = \frac{\dot{v}_{\infty,c}}{g} - \gamma_c \quad (12.14)$$

which make up the reference signal to track, i.e.

$$\vec{r} = \begin{bmatrix} \dot{\tilde{\mathcal{E}}}_c \\ \dot{\tilde{\mathcal{L}}}_c \end{bmatrix} = \begin{bmatrix} \frac{\dot{v}_{\infty,c}}{g} + \gamma_c \\ \frac{\dot{v}_{\infty,c}}{g} - \gamma_c \end{bmatrix} \quad (12.15)$$

with output

$$\vec{y} = \begin{bmatrix} \dot{\tilde{\mathcal{E}}} \\ \dot{\tilde{\mathcal{L}}} \end{bmatrix} = \begin{bmatrix} \frac{\dot{v}_{\infty}}{g} + \gamma \\ \frac{\dot{v}_{\infty}}{g} - \gamma \end{bmatrix} \quad (12.16)$$

with tracking error

$$\vec{e} = \vec{y} - \vec{r} \quad (12.17)$$

Next, consider that one desires to track constant step inputs to \vec{r} and reject constant disturbances, \vec{w} , e.g. the additive drag, one can use the servomechanism state-space augmentation with $p = 1$ and $a_1 = 0$ as

$$\vec{z} = \begin{bmatrix} \vec{e} \\ \vec{y} \end{bmatrix} \quad (12.18)$$

and

$$\vec{\mu} = \dot{\vec{u}} \quad (12.19)$$

where one can design a feedback control law on the servomechanism as

$$\vec{\mu} = K_z \vec{z} \quad (12.20)$$

or

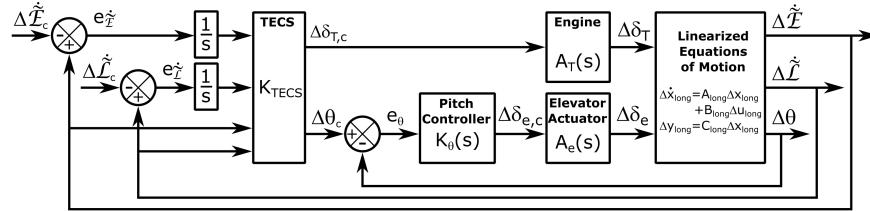
$$\vec{u} = [K_e \quad K_y] \begin{bmatrix} \int \vec{e} \\ \vec{y} \end{bmatrix} \quad (12.21)$$

where $K_e \in \mathbb{R}^{n_u \times n_y}$ and $K_y \in \mathbb{R}^{n_u \times n_y}$.

For the TECS, one simplifies this general structure into four non-zero and four zero elements of the gain matrix, i.e.

$$\begin{bmatrix} \delta_T \\ \theta_c \end{bmatrix} = \begin{bmatrix} K_{e,\dot{\tilde{\mathcal{E}}}} & 0 & K_{y,\dot{\tilde{\mathcal{E}}}} & 0 \\ 0 & K_{e,\dot{\tilde{\mathcal{L}}}} & 0 & K_{y,\dot{\tilde{\mathcal{L}}}} \end{bmatrix} \begin{bmatrix} \int (\dot{\tilde{\mathcal{E}}} - \dot{\tilde{\mathcal{E}}}_c) \\ \int (\dot{\tilde{\mathcal{L}}} - \dot{\tilde{\mathcal{L}}}_c) \\ \dot{\tilde{\mathcal{E}}} \\ \dot{\tilde{\mathcal{L}}} \end{bmatrix} = K_{TECS} \begin{bmatrix} \int (\dot{\tilde{\mathcal{E}}}_c - \dot{\tilde{\mathcal{E}}}) \\ \int (\dot{\tilde{\mathcal{L}}}_c - \dot{\tilde{\mathcal{L}}}) \\ \dot{\tilde{\mathcal{E}}} \\ \dot{\tilde{\mathcal{L}}} \end{bmatrix} \quad (12.22)$$

Thus, a TECS uses the two proportional gains, $K_{y,\dot{\mathcal{E}}}$ and $K_{y,\dot{\mathcal{L}}}$, to stabilize the system dynamics and two integral gains, $K_{e,\dot{\mathcal{E}}}$ and $K_{e,\dot{\mathcal{L}}}$, to reject disturbances and achieve zero steady-state error to step commands. These gains as the matrix, \mathbf{K}_z , can be designed using linear control methods, i.e. eigenvalue placement or optimal control. Thus, for control design, one can model a TECS system as the block diagram



where $\Delta\dot{\mathcal{E}}_c$ and $\Delta\dot{\mathcal{L}}_c$ can be computed directly from $\Delta\gamma_c$ and $\Delta v_{\infty,c}$ and $\Delta\dot{\mathcal{E}}$ and $\Delta\dot{\mathcal{L}}$ could also be computed instead.

The θ_c is passed to a SISO inner-loop controller for $\delta_{e,c}$, as

$$\delta_{e,c}(s) = K_\theta(s)\theta_c(s) \quad (12.23)$$

which is typically designed first to track the θ_c signal faster than the outer TECS loop and improve the stability characteristics of the short-period dynamics. For control design of the TECS outer-loop, the linearized plant will correspond to some LTI state-space model as

$$\begin{aligned} \dot{\vec{x}}_{cl} &= A_{cl}\vec{x}_{cl} + B_{cl}\vec{u} \\ \vec{y} &= C_{cl}\vec{x}_{cl} + D_{cl}\vec{u} \end{aligned} \quad (12.24)$$

where \vec{x}_{cl} includes the state of the vehicle, elevator actuator, engine, and controller, $\vec{u} = [\Delta\delta_T \ \Delta\theta_c]^T$ and $\vec{y} = [\dot{\mathcal{E}} \ \dot{\mathcal{L}}]^T$.

Next, defining the augmented state as

$$\vec{x}_{aug} = \begin{bmatrix} \int e_{\dot{\mathcal{E}}} \\ \int e_{\dot{\mathcal{L}}} \\ \vec{x}_{cl} \end{bmatrix} \quad (12.25)$$

with output feedback

$$\vec{u} = K_{TECS} (C_{aug}\vec{x}_{aug} + D_{aug}\vec{u}_{aug}) \quad (12.26)$$

one has closed-loop dynamics

$$\begin{aligned} \dot{\vec{x}}_{aug} &= \left(A_{aug} + B_{aug}K_{TECS}(I_2 + D_{aug}K_{TECS})^{-1}C_{aug} \right) \vec{x}_{aug} + \begin{bmatrix} -I_2 \\ 0 \end{bmatrix} \vec{r} \\ \vec{y}_{aug} &= C_{aug}\vec{x}_{aug} \end{aligned} \quad (12.27)$$

where

$$A_{aug} = \begin{bmatrix} 0 & C_{cl} \\ 0 & A_{cl} \end{bmatrix} \quad (12.28)$$

$$B_{aug} = \begin{bmatrix} D_{cl} \\ B_{cl} \end{bmatrix} \quad (12.29)$$

$$C_{aug} = \begin{bmatrix} I_2 & 0 \\ 0 & C_{cl} \end{bmatrix} \quad (12.30)$$

and

$$D_{aug} = \begin{bmatrix} 0 \\ D_{cl} \end{bmatrix} \quad (12.31)$$

Total Heading Control System

Analogously to the TECS coupled commands and error rates of γ and $\frac{\dot{v}_\infty}{g}$ for efficient decoupling of the thrust and elevator, a **total heading control system (THCS)** couples the yaw angle, ψ , with the sideslip angle, β , to efficiently control the flight vehicle via the error rates of the heading, σ , i.e. the $\psi + \beta$, as well as the differential $\psi - \beta$, in order to form a commanded roll angle and commanded yaw rate to the inner-loop controllers. In particular, one computes the commanded yaw rate as

$$\dot{\psi}_c = K_\psi(\psi_c - \psi) \quad (12.32)$$

and the commanded sideslip as

$$\dot{\beta}_c = K_\beta(\beta_c - \beta) \quad (12.33)$$

where typically these are also limited using some minimum and maximum values for the rate commands.

Then, using defining the error rates as

$$e_\psi = \dot{\psi}_c - \dot{\psi} \quad (12.34)$$

and

$$e_\beta = \dot{\beta}_c - \dot{\beta} \quad (12.35)$$

one can form the commanded roll angle as

$$\phi_c = \frac{K_{\phi,i}}{s} \frac{v_\infty}{g} (e_\psi + e_\beta) \quad (12.36)$$

and the commanded yaw rate angle as

$$r_c = \frac{K_{r,i}}{s} (e_\psi - e_\beta) \quad (12.37)$$

where the $\frac{v_\infty}{g}$ term is used to coordinate r_c with ϕ_c . Furthermore, for coordinated turns one requires $K_\psi = K_\beta$ and for $\phi_c = \frac{v_\infty r_c}{g}$, one requires $K_{\phi,i} = K_{r,i}$.

In this way, one can further design the control gains for the decoupled inner-loops, i.e.

$$\delta_{a,c} = K_p f(\bar{q})(K_\phi(\phi_c - \phi) - p) \quad (12.38)$$

and

$$\delta_{r,c} = K_r f(\bar{q})(r_c - r) \quad (12.39)$$

to provide stability augmentation for all control modes and flight conditions for the specific flight vehicle. Here $f(\bar{q})$ is a multiplicative “inversion” function that is multiplied by the control gains and is dependent on the trim pitch rate, \bar{q} . Alternatively, one could implement a dynamics inversion block instead of a multiplicative function.

12.3 Rotary-Wing Attitude Control Systems

Traditional Helicopter Attitude Control Systems

Multi-Rotor Attitude Control Systems

12.4 Spacecraft Four-Wheel Attitude Control Systems

Reaction Thruster Attitude Control

Magnetic Attitude Control

Momentum Dumping

12.5 Spacecraft Attitude-Maneuver Systems

Time-Optimal Attitude-Maneuvers

Despinning Maneuver

Denutation Maneuver

Coning Maneuver

$\Delta\vec{H}_G$ are used to perform **coning maneuvers** to reorient the spin axis by some angle θ_c . Consider a spinning satellite with some initial $\vec{H}_{G,0}$ about the spin axis, z_B . A thruster pair impulsively fires creating some $\Delta H_{G,1}$ normal to the spin axis. This induces a precession, or coning, of the satellite about an axis at an angle $\theta_c/2$ to $H_{G,0}$. Following the torque-free motion model, the precession rate after the impulse is given by

$$\omega_p = \frac{I_3 \omega_s}{I_1 - I_3} \sec\left(\frac{\theta_c}{2}\right) \quad (12.40)$$

Then, after precessing π radians, the thruster pair impulsively fires creating some $\Delta H_{G,2}$ in the same direction relative to the spacecraft as $\Delta H_{G,1}$ with $\|\Delta\vec{H}_{G,2}\|_2 = \|\Delta\vec{H}_{G,1}\|_2$, stabilizing the spin vector in the commanded reorientation, θ . The total ΔH for the single coning maneuver is

$$\Delta H_{tot} = \|\Delta\vec{H}_{G,1}\|_2 + \|\Delta\vec{H}_{G,2}\|_2 = 2\|\vec{H}_{G,0}\|_2 \tan\left(\frac{\theta_c}{2}\right) \quad (12.41)$$

The time required for this single coning maneuver, t_1 , is found by dividing the precession angle by the rate, i.e.

$$t_1 = \frac{\pi}{\omega_p} = \pi \frac{I_1 - I_3}{I_3 \omega_s} \cos\left(\frac{\theta_c}{2}\right) \quad (12.42)$$

However, as propellant expenditure is reflected in the magnitude of the individual angular momentum increments, one can reduce the amount of fully expended fuel by a sequence of N small coning maneuvers rather than one large maneuver. For these N small coning maneuvers, one has

$$\Delta H_{tot} = 2N\|\vec{H}_{G,0}\|_2 \tan\left(\frac{\theta_c}{2N}\right) \quad (12.43)$$

However, for large N , one has by the small angle approximation

$$\Delta H_{tot} \approx 2N\|\vec{H}_{G,0}\|_2 \left(\frac{\theta_c}{2N} \right) \approx \|\vec{H}_{G,0}\|_2 \theta_c \quad (12.44)$$

However, the time is increased to perform the reorientation, i.e.

$$t_N = N\pi \frac{I_1 - I_3}{I_3 \omega_s} \cos \left(\frac{\theta_c}{2N} \right) \quad (12.45)$$

or in ratio form, one has by the small angle approximation

$$\frac{t_N}{t_1} = N \frac{\cos \left(\frac{\theta_c}{2N} \right)}{\cos \left(\frac{\theta_c}{2} \right)} \approx \frac{N}{\cos \left(\frac{\theta_c}{2} \right)} \quad (12.46)$$

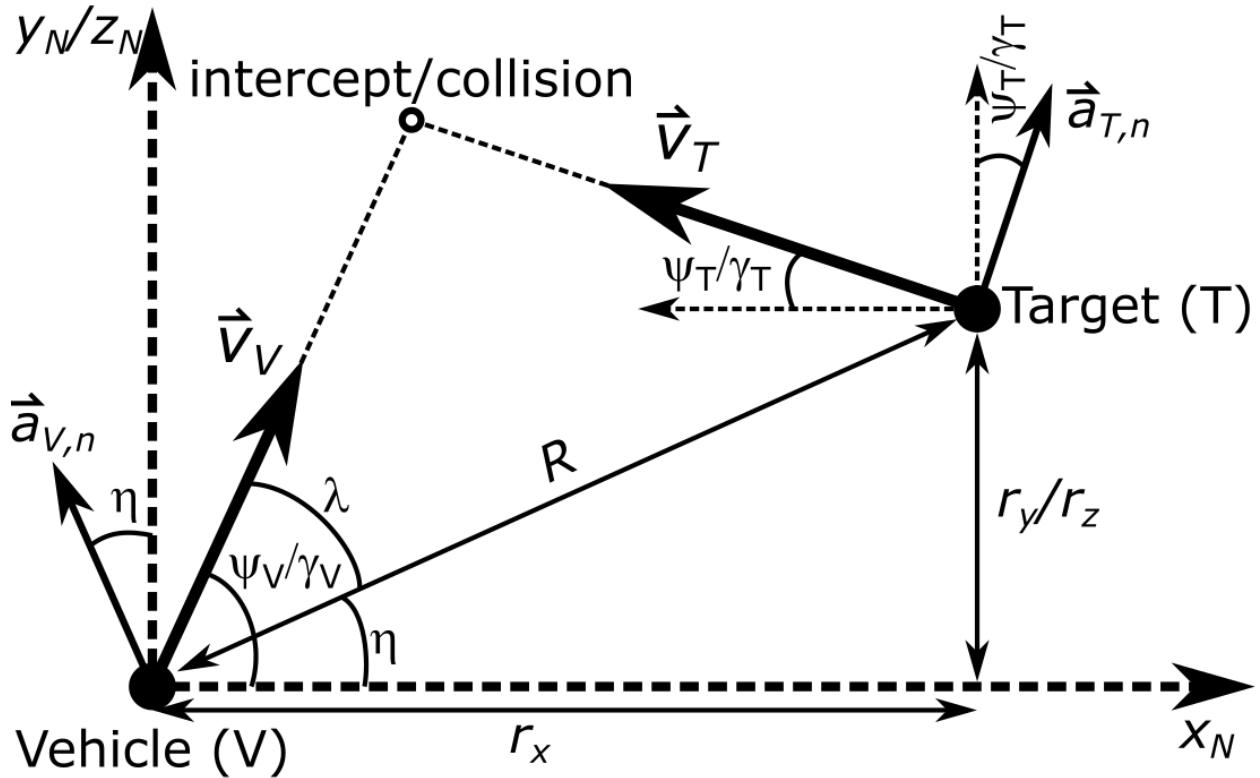
References

For more information, please refer to the following

- Curtis, H. D., “10.6 Coning Maneuver,” *Orbital Mechanics for Engineering Students*, 1st ed., Vol. 1, Elsevier Butterworth-Heinemann, Massachusetts, 2005, pp. 503-506
-

12.6 Planar Intercept Guidance Systems

While flight vehicles operate in three-dimensional space, many, if not most, guidance laws devise and implement planar guidance laws in each of the two maneuver planes, e.g. longitudinal and lateral-directional. This approach also includes the **intercept guidance problem**, also known as **pursuit-evasion guidance problem**. This problem is depicted in the following diagram



where ψ_V/γ_V is the heading/flight-path angle of the vehicle, ψ_T/γ_T is the heading/flight-path angle of the target, \vec{v}_V is the velocity of the vehicle, \vec{v}_T is the velocity of the target, r_x is the x_N relative position, r_y/r_z is the relative y_N/z_N position, R is the range-to-target, η is the **line-of-sight (LOS) angle**, $\vec{a}_{V,n}$ is the acceleration of the vehicle normal to the LOS, $\vec{a}_{T,n}$ is the acceleration of the target normal to its velocity, and λ is the **lead angle** between the LOS vector and the vehicle velocity vector which must be positive and $\dot{R} < 0$ for an intercept to exist. Determining this lead angle is a key purpose of the guidance law. In this case, if the vehicle actively reduces and holds r_y/r_z to zero, then r_x will continue to decrease until collision occurs. Thus, regulating r_y/r_z is the key analysis required for planar intercept.

In Cartesian form, the intercept kinematics can be stated as the target-vehicle relative position, \vec{r} ,

$$\vec{r} = \vec{r}_T - \vec{r}_V = \begin{bmatrix} r_x \\ r_y \\ r_z \end{bmatrix} = \begin{bmatrix} R \cos \eta \\ R \sin \eta \\ 0 \end{bmatrix} \quad (12.47)$$

the relative velocity

$$\vec{v} = \begin{bmatrix} v_x \\ v_y \\ v_z \end{bmatrix} = \begin{bmatrix} -\|\vec{v}_T\|_2 \cos \gamma_T - \|\vec{v}_V\|_2 \cos(\lambda + \eta) \\ \|\vec{v}_T\|_2 \sin \gamma_T - \|\vec{v}_V\|_2 \sin(\lambda + \eta) \\ 0 \end{bmatrix} \quad (12.48)$$

and the relative acceleration

$$\vec{a} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} = \begin{bmatrix} \|\vec{a}_{T,n}\|_2 \sin \gamma_T + \|\vec{a}_{V,n}\|_2 \sin \eta \\ \|\vec{a}_{T,n}\|_2 \cos \gamma_T - \|\vec{a}_{V,n}\|_2 \cos \eta \\ 0 \end{bmatrix} \quad (12.49)$$

These are clearly nonlinear equations.

However, with small angle approximations for γ_T and η , i.e. near-collision course conditions, one has

$$r_y \approx R\eta \quad (12.50)$$

and

$$a_y \approx a_{T,n} - a_{V,n} \quad (12.51)$$

which allows the kinematics

$$v_y = \int \| \vec{a}_{T,n} \|_2 - \| \vec{a}_{V,n} \|_2 \quad (12.52)$$

and

$$r_y = \int \int \| \vec{a}_{T,n} \|_2 - \| \vec{a}_{V,n} \|_2 \quad (12.53)$$

which can be written as an augmented LTI state-space system

$$\begin{bmatrix} \dot{r}_y \\ \dot{v}_y \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} r_y \\ v_y \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \| \vec{a}_{V,n} \|_2 + \begin{bmatrix} 0 \\ 1 \end{bmatrix} \| \vec{a}_{T,n} \|_2 \quad (12.54)$$

$$\vec{y} = \begin{bmatrix} r_y \\ v_y \end{bmatrix}$$

which can be used to form different linear-quadratic guidance laws depending on what additional assumptions are made. Note that if r_y and v_y are not both available for feedback, then one can employ the separation principle and use a parallel guidance filter to estimate r_y and v_y .

Constant-Velocity Guidance Laws

Assuming the vehicle and target speeds, $\| \vec{v}_V \|_2$ and $\| \vec{v}_T \|_2$, are constant, the target is non-manuevering, i.e. $\| \vec{a}_{T,n} \|_2 = 0$, and the vehicle responds instantaneously to an acceleration command, $a_{V,c} = \| \vec{a}_{V,n} \|_2$, one can form the following LQ minimum-energy OCP as

$$\begin{aligned} \vec{u}^{\text{opt}}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\text{argmin}} \quad \mathcal{J} = \vec{x}^T(t_f) \begin{bmatrix} E_r & 0 \\ 0 & E_v \end{bmatrix} \vec{x}(t_f) + \int_0^{t_f} \vec{u}^T(t) \vec{u}(t) dt \\ \text{subject to: } \dot{\vec{x}}(t) &= \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 \\ -1 \end{bmatrix} \vec{u}(t) \\ \text{initial condition: } \vec{x}(0) & \end{aligned} \quad (12.55)$$

where $\vec{x} = [r_y \ v_y]^T$, $u = a_{V,c}$, E_r is the cost on the relative range at time t_f , and E_v is the cost on the relative velocity at time t_f . When $E_r \rightarrow \infty$ and $E_v = 0$, one has an **intercept problem** and when $E_r \rightarrow \infty$ and $E_v \rightarrow \infty$, one has an **rendezvous problem**.

The solution to this OCP is given by the differential Riccati equation

$$\dot{P} = -PA - A^T P + PBB^T P \quad (12.56)$$

with endpoint condition

$$P(t_f) = \begin{bmatrix} E_r & 0 \\ 0 & E_v \end{bmatrix} \quad (12.57)$$

where the optimal control is given by

$$u^{opt}(t) = -B^T P(t) \vec{x} \quad (12.58)$$

which can be solved as

$$u^{opt}(t) = \frac{3}{t_{go}^2} \left[\frac{\left(1 + \frac{1}{2}E_v t_{go}\right) r_y(t) + \left(1 + \frac{1}{3}E_v t_{go} + \frac{E_v}{E_r t_{go}^2}\right) v_y(t) t_{go}}{1 + \frac{3}{E_r t_{go}^3} (1 + E_v t_{go}) + \frac{E_v t_{go}}{4}} \right] \quad (12.59)$$

where $t_{go} = t_f - t$.

This can be expressed as the following structure

$$u^{opt}(t) = \frac{\tilde{N}(E_r, E_v, t_{go})}{t_{go}^2} Z(r_y, v_y, E_r, E_v, t_{go}) \quad (12.60)$$

with

$$\tilde{N}(E_r, E_v, t_{go}) = \frac{3}{1 + \frac{3}{E_r t_{go}^3} (1 + E_v t_{go}) + \frac{E_v t_{go}}{4}} \quad (12.61)$$

as the **effective navigation ratio** and

$$Z(r_y, v_y, E_r, E_v, t_{go}) = \left(1 + \frac{1}{2}E_v t_{go}\right) r_y(t) + \left(1 + \frac{1}{3}E_v t_{go} + \frac{E_v}{E_r t_{go}^2}\right) v_y(t) t_{go} \quad (12.62)$$

The **proportional guidance (PN) guidance** occurs for the intercept problem, i.e. if $E_v = 0$ and as $E_r \rightarrow \infty$, one has

$$u_{PN}(t) = \frac{3}{t_{go}^2} [r_y(t) + v_y(t) t_{go}] \quad (12.63)$$

where $\tilde{N}_{PN} = 3$ and $Z = ZEM_{PN} = r_y(t) + v_y(t) t_{go}$ which is often referred to as the **zero-effort-miss**, i.e. the current miss distance that would result if the vehicle and the target did not maneuver over the time period $[t, t_f]$, which, in practice, is estimated and fed to the controller by a guidance filter. Likewise, the **rendezvous (REN) guidance** law occurs for the rendezvous problem, i.e. as $E_v, E_r \rightarrow \infty$, one has

$$u_{REN}(t) = \frac{6}{t_{go}^2} \left[r_y(t) + \frac{2}{3} v_y(t) t_{go} \right] \quad (12.64)$$

As an aside, recall that

$$r_y \approx R\lambda \quad (12.65)$$

$$v_y = \dot{R}\lambda + R\dot{\lambda} \quad (12.66)$$

where $R = -\dot{R}t_{go}$. Thus, substituting for λ and R , one has

$$v_y = \dot{R} \frac{r_y}{-\dot{R}t_{go}} - \dot{R}t_{go}\dot{\lambda} \quad (12.67)$$

and rearranging, one has

$$\dot{R}\lambda = \frac{r_y(t) + v_y(t)t_{go}}{t_{go}^2} \quad (12.68)$$

and the PN guidance law can be alternatively expressed as

$$u_{PN}(t) = -\tilde{N}\dot{R}\dot{\eta} \quad (12.69)$$

where $\tilde{N} = 3$ is the “energy-optimal” PN gain as shown previously.

Constant-Acceleration Guidance Law

Next, assume that the target is undergoing a constant acceleration, $\|\vec{a}_{T,n}\|_2 = a_{T,n,y}$, which can be added to the state and estimated. Then, one can form the following LQ minimum-energy OCP as

$$\begin{aligned} \vec{u}^{opt}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\operatorname{argmin}} \quad \mathcal{J} = \vec{x}^T(t_f) \begin{bmatrix} E_r & 0 & 0 \\ 0 & E_v & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{x}(t_f) + \int_0^{t_f} \vec{u}^T(t) \vec{u}(t) dt \\ \text{subject to: } \dot{\vec{x}}(t) &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 \\ -1 \\ 0 \end{bmatrix} \vec{u}(t) \\ \text{initial condition: } \vec{x}(0) & \end{aligned} \quad (12.70)$$

where $\vec{x} = [r_y \ v_y \ a_{T,n,y}]^T$ and $u = \|\vec{a}_{V,n}\|_2$.

This provides the following general solution

$$u^{opt}(t) = \frac{3}{t_{go}^2} \left[\frac{\left(1 + \frac{1}{2}E_v t_{go}\right)r_y(t) + \left(1 + \frac{1}{3}E_v t_{go} + \frac{E_v}{E_r t_{go}^2}\right)v_y(t)t_{go} + \frac{1}{2}\left(1 + \frac{1}{6}E_v t_{go} + \frac{2E_v}{E_r t_{go}^2}\right)a_{T,n,y}(t)t_{go}^2}{1 + \frac{3}{E_r t_{go}^3}(1 + E_v t_{go}) + \frac{E_v t_{go}}{4}} \right] \quad (12.71)$$

which show that the only difference is the additional term in the numerator which is time-varying gain multiplying the target acceleration state.

The **augmented proportional guidance (APN) guidance** occurs for the intercept problem, i.e. if $E_v = 0$ and as $E_r \rightarrow \infty$, one has

$$u_{APN}(t) = \frac{3}{t_{go}^2} \left[r_y(t) + v_y(t)t_{go} + \frac{1}{2}a_{T,n,y}t_{go}^2 \right] \quad (12.72)$$

where $\tilde{N}_{APN} = 3$ and $Z = ZEM_{APN} = r_y(t) + v_y(t)t_{go} + \frac{1}{2}a_{T,n,y}t_{go}^2$. Thus, the only change in the APN from PN is a change in the ZEM estimate. As such, the APN guidance law will only perform better than PN if the vehicle is able to sufficiently estimate the target normal acceleration.

Similarly, the **augmented rendezvous (AREN) guidance** law occurs for the rendezvous problem, i.e. as $E_v, E_r \rightarrow \infty$, one has

$$u_{AREN}(t) = \frac{6}{t_{go}^2} \left[r_y(t) + \frac{2}{3}v_y(t)t_{go} \right] + a_{T,n,y} \quad (12.73)$$

which is the same as REN, but adds a direct cancellation of the target maneuver in the acceleration command.

Furthermore, consider the case where the vehicle does not respond immediately to an acceleration command, but experiences some lag modeled as a first-order transfer function

$$\frac{\|\vec{a}_{V,n}\|_2}{a_{V,c}} = \frac{\omega}{s + \omega} \quad (12.74)$$

Then, one can form the following LQ minimum-energy OCP as

$$\begin{aligned} \vec{u}^{\text{opt}}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\text{argmin}} \quad \mathcal{J} = \vec{x}^T(t_f) \begin{bmatrix} E_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}(t_f) + \int_0^{t_f} \vec{u}^T(t) \vec{u}(t) dt \\ \text{subject to: } \dot{\vec{x}}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -\omega \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 \\ 0 \\ 0 \\ \omega \end{bmatrix} \vec{u}(t) \quad (12.75) \\ \text{initial condition: } \vec{x}(0) & \end{aligned}$$

where $\vec{x} = [r_y \ v_y \ a_{T,n,y} \ \|\vec{a}_{V,n}\|_2]^T$ and $u = a_{V,c}$.

This provides the following general solution

$$u^{opt}(t) = \frac{6\omega^2 t_{go}^2 (\omega t_{go} + e^{-\omega t_{go}} - 1)}{t_{go}^2} \left[\frac{r_y(t) + v_y(t)t_{go} + \frac{1}{2}a_{T,n,y}(t)t_{go}^2 - \frac{1}{\omega^2}(\omega t_{go} + e^{-\omega t_{go}} - 1)\|\vec{a}_{V,n}\|_2}{\frac{6\omega^3}{E_r} + 3 + 6\omega t_{go} - 6\omega^2 t_{go}^2 + 2\omega^3 t_{go}^3 - 12\omega t_{go}e^{-\omega t_{go}} - 3e^{-2\omega t_{go}}} \right] \quad (12.76)$$

which is considerably more complex than the non-ideal vehicle response to an acceleration command. The **optimal guidance law (OGL)** occurs for the intercept problem, i.e. as $E_v \rightarrow \infty$, one has

$$u_{OGL}(t) = \frac{6\omega^2 t_{go}^2 (\omega t_{go} + e^{-\omega t_{go}} - 1)}{t_{go}^2} \left[\frac{r_y(t) + v_y(t)t_{go} - \frac{1}{\omega^2}(\omega t_{go} + e^{-\omega t_{go}} - 1)\|\vec{a}_{V,n}\|_2}{3 + 6\omega t_{go} - 6\omega^2 t_{go}^2 + 2\omega^3 t_{go}^3 - 12\omega t_{go}e^{-\omega t_{go}} - 3e^{-2\omega t_{go}}} \right] \quad (12.77)$$

where the time-varying effective navigation ratio is

$$\tilde{N}_{OGL} = \frac{6\omega^2 t_{go}^2 (\omega t_{go} + e^{-\omega t_{go}} - 1)}{3 + 6\omega t_{go} - 6\omega^2 t_{go}^2 + 2\omega^3 t_{go}^3 - 12\omega t_{go}e^{-\omega t_{go}} - 3e^{-2\omega t_{go}}} \quad (12.78)$$

and zero-effort-miss is

$$ZEM_{OGL} = r_y(t) + v_y(t)t_{go} - \frac{1}{\omega^2}(\omega t_{go} + e^{-\omega t_{go}} - 1)\|\vec{a}_{V,n}\|_2 \quad (12.79)$$

which simply adds another term to the ZEM_{APN} based on the response lag.

Constant-Jerk Guidance Laws

Finally, assume that the target is undergoing a constant jerk, $j_{T,n,y}$, which can be added to the state along with the target acceleration and estimated. Then, one can form the following LQ minimum-energy OCP as

$$\begin{aligned} \vec{u}^{\text{opt}}(t) &= \underset{u(t) \forall t \in [0, t_f]}{\text{argmin}} \quad \mathcal{J} = \vec{x}^T(t_f) \begin{bmatrix} E_r & 0 & 0 & 0 \\ 0 & E_v & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}(t_f) + \int_0^{t_f} \vec{u}^T(t) \vec{u}(t) dt \\ \text{subject to: } \dot{\vec{x}}(t) &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix} \vec{x}(t) + \begin{bmatrix} 0 \\ -1 \\ 0 \\ 0 \end{bmatrix} \vec{u}(t) \\ \text{initial condition: } \vec{x}(0) & \end{aligned} \quad (12.80)$$

where $\vec{x} = [r_y \ v_y \ a_{T,n,y} \ j_{T,n,y}]^T$ and $u = \|\vec{a}_{V,n}\|_2$.

This provides the following general solution

$$u^{\text{opt}}(t) = \frac{3}{t_{go}^2} \left[\frac{\left(1 + \frac{1}{2}E_v t_{go}\right)r_y(t) + \left(1 + \frac{1}{3}E_v t_{go} + \frac{E_v}{E_r t_{go}^2}\right)v_y(t)t_{go} + \frac{1}{2}\left(1 + \frac{1}{6}E_v t_{go} + \frac{2E_v}{E_r t_{go}^2}\right)a_{T,n,y}(t)t_{go}^2 + \frac{1}{6}\left(1 + \frac{3E_v}{E_r t_{go}^2}\right)j_{T,n,y}(t)t_{go}^3}{1 + \frac{3}{E_r t_{go}^3}(1 + E_v t_{go}) + \frac{E_v t_{go}}{4}} \right] \quad (12.81)$$

which show that the only difference is the additional term in the numerator which is time-varying gain multiplying the target acceleration state.

The **extended proportional guidance (EPN) guidance** occurs for the intercept problem, i.e. if $E_v = 0$ and as $E_r \rightarrow \infty$, one has

$$u_{EPN}(t) = \frac{3}{t_{go}^2} \left[r_y(t) + v_y(t)t_{go} + \frac{1}{2}a_{T,n,y}t_{go}^2 + \frac{1}{6}j_{T,n,y}(t)t_{go}^3 \right] \quad (12.82)$$

where $\tilde{N}_{EPN} = 3$ and $Z = ZEM_{EPN} = r_y(t) + v_y(t)t_{go} + \frac{1}{2}a_{T,n,y}t_{go}^2 + \frac{1}{6}j_{T,n,y}(t)t_{go}^3$. Thus, the only change in the EPN from APN and PN is a change in the ZEM estimate. As such, the EPN guidance law will only perform better if the vehicle is able to sufficiently estimate the target normal acceleration *and* its jerk.

12.7 L_1 Guidance Systems

12.8 Powered Explicit Guidance Systems

Powered Descent Guidance (PDG)

E-Guidance (EG)

Apollo Powered Guidance (APG)

3-DoF PDG

6-DoF PDG

Part III

Probability and Estimation Theory for Perception

Probability Theory for Perception

13.1 Probability Theory and Random Variables

This chapter presents an overview of important concepts from probability theory for the purposes of perception systems. For brevity, it is devoid of proofs for the axioms, theorems, and concepts presented. The interested reader is encouraged to use the reference materials and other resources for more in-depth studies of probability theory which will provide further insight into the design and analysis of aerospace vehicle perception systems beyond this textbook. This chapter also introduces more set theory definitions which are reviewed in section A.1 in the appendix.

Frequentist Probability Theory

One justification of probability is based on **frequentist probability theory**. Here one defines the **outcomes** as all possible results of a “yet-to-be-performed” experiment. The probabilities of occurrence can be assigned to sets of its potential values known as **events**. To assign probabilities to events from a frequentist viewpoint, one would perform an experiment N times with some number of possible outcomes corresponding to events E_1, E_2, \dots , count the number of times each event occurred, i.e., N_1, N_2, \dots . Then, forming ratios of the number of occurrences of each events, i.e., the **relative frequencies**, one has

$$\frac{N_1}{N}, \frac{N_2}{N}, \dots \quad (13.1)$$

Then, a frequentist would assign the probabilities of each event based on the concept of **statistical regularity** in that these relative frequencies converge to some limiting value which provides the basis for the frequentist definition of probability as

$$\mathbb{P}(E_1) = \lim_{N \rightarrow \infty} \frac{N_1}{N}, \mathbb{P}(E_2) = \lim_{N \rightarrow \infty} \frac{N_2}{N}, \dots \quad (13.2)$$

As an example, consider rolling two dice. This experiment would have the following outcomes:

$$\begin{aligned}
 & (1, 1) \\
 & (1, \cdot) \\
 & (1, 6) \\
 & (2, 1) \\
 & (2, \cdot) \\
 & (2, 6) \\
 & (\cdot, \cdot) \\
 & (6, 6)
 \end{aligned} \tag{13.3}$$

which mathematically provides thirty-six possible outcomes for this experiment. Considering the random variable as the sum of the two dice, and the event E_1 would be the sum of two dice being equal to 2, which has one outcome. Similarly, the event E_2 could be assigned as the sum of dice begin equal to 3 which has two outcomes. This could be continued for all eleven possible events. Note that if the two dice are **fair**, i.e., each outcome is equally probable, then $\mathbb{P}(E_1) = \frac{1}{36}$, $\mathbb{P}(E_2) = \frac{2}{36} = \frac{1}{18}$, \dots .

Axiomatic Probability Theory

Another justification of probability is Kolmogorov's **axiomatic probability theory** which relies on set theory and measure theory. In this theory, all possible outcomes of some non-empty phenomenon make up the set Ω known as the **outcome space**, also known as the **possibility space** or the **sample space**. All events make up the **event space**, \mathcal{E} . The set function, \mathbb{P} , is a measure that returns any event's probability, i.e., it satisfies the first axiom

$$\mathbb{P}(E) \geq 0 \in \mathbb{R} \quad \forall E \in \mathcal{E} \tag{13.4}$$

with unit measure as the second axiom

$$\mathbb{P}(\Omega) = 1 \tag{13.5}$$

and for any mutually exclusive or pairwise disjoint events, E_1, E_2, \dots , it also satisfies the third axiom

$$\mathbb{P}\left(\bigcup_{i=1}^{\infty} E_i\right) = \sum_{i=1}^{\infty} \mathbb{P}(E_i) \tag{13.6}$$

which is known as **σ -additivity**.

From these three axioms, one can directly infer the following five corollaries for the probability of the empty set as

$$\mathbb{P}(\emptyset) = 0 \tag{13.7}$$

the probability of finite disjoint unions

$$\mathbb{P}\left(\bigcup_{i=1}^n E_i\right) = \sum_{i=1}^n \mathbb{P}(E_i) \tag{13.8}$$

the probability of the complement as

$$\mathbb{P}(E^c) = 1 - \mathbb{P}(E) \quad (13.9)$$

the monotonicity of \mathbb{P} as

$$\text{if } E_1 \subseteq E_2 \text{ then } \mathbb{P}(E_1) \leq \mathbb{P}(E_2) \quad (13.10)$$

and the **addition law of probability** as

$$\mathbb{P}(E_1 \cup E_2) = \mathbb{P}(E_1) + \mathbb{P}(E_2) - \mathbb{P}(E_1 \cap E_2) \quad (13.11)$$

where $\mathbb{P}(E_1 \cap E_2) = 0$ if events E_1 and E_2 are **mutually exclusive**. This final corollary can be extended to any number of sets as the **inclusion-exclusion principle**

$$\mathbb{P}\left(\bigcup_{i=1}^n E_i\right) = \sum_{k=1}^n \left((-1)^{k-1} \sum_{I \subseteq \{1, \dots, n\}, |I|=k} \mathbb{P}\left(\bigcap_{i \in I} E_i\right) \right) \quad (13.12)$$

where the last sum iterates over all subsets I of the indices $1, \dots, n$ which contain exactly k elements. It can also be noted that any **measure** is defined by axioms 1 and 3 and corollary 1 while using axiom 2 instead of corollary 1 is specific to probability measures.

In set theory, the use of these three axioms and their corollaries infer that the ordered triplet $(\Omega, \mathcal{E}, \mathbb{P})$ forms a **probability space** and the collection, \mathcal{A} , of all possible events $E_i \in \mathcal{E}$ is a σ -algebra on Ω , i.e., it contains the empty set and is closed under complements and countable unions. The formulation of this definition is important for formalizing probability measures for uncountably infinite sample spaces, Ω , where one must “construct” a countably infinite collection of events, e.g., the real number line which is done through the use of Borel sets. A **Borel set** is any set in a topological space that can be formed from open sets through the operations of countable union, countable intersection, and relative complement. For certain Ω , the collection of all Borel sets on Ω forms a Borel σ -algebra on Ω which is the smallest σ -algebra containing all open sets, e.g., the Borel σ -algebra on \mathbb{R} is the smallest σ -algebra on \mathbb{R} that contains all possible *intervals* of the real numbers. Lastly, it should be noted that \mathbb{P} is defined only for events and need not be defined for outcomes. However, for countable outcome spaces Ω , one can define an **elementary event**, also known as a **sample point** as an event that contains only a single outcome. Thus, allowing outcomes to be assigned probabilities indirectly for countable Ω . Further details on this measure-theoretic basis for formal definitions of probability, especially for continuous random variables, is beyond the scope of this textbook, but is mentioned to provide readers with some exposure to formal ways of understanding the peculiarities of probability modeling.

Fundamental Concepts in Probability Theory for Perception

An important concept in **conditional probability** is defined as the probability that the event E_1 occurs given that event E_2 has occurred which is denoted as

$$\mathbb{P}(E_1|E_2) = \frac{\mathbb{P}(E_1 \cap E_2)}{\mathbb{P}(E_2)} \text{ if } \mathbb{P}(E_2) \neq 0 \quad (13.13)$$

This can also be rearranged as

$$\mathbb{P}(E_1 \cap E_2) = \mathbb{P}(E_1|E_2)\mathbb{P}(E_2) \quad (13.14)$$

Then, considering the conditional probability of E_2 given E_1 has occurred, one obtains **Bayes' theorem**, also known as **Bayes' law** or **Bayes' rule**, for two events as

$$\mathbb{P}(E_2|E_1) = \frac{\mathbb{P}(E_1|E_2)\mathbb{P}(E_2)}{\mathbb{P}(E_1)} \quad (13.15)$$

where $\mathbb{P}(E_2)$ is called the **prior probability**, $\mathbb{P}(E_2|E_1)$ is known as the **posterior probability**, $\mathbb{P}(E_1|E_2)$ is known as the **likelihood**. Notably the quotient $\mathbb{P}(E_1|E_2)/\mathbb{P}(E_1)$ is referred to as the **support** event E_1 provides for event E_2 .

Another important concept is to consider the event E_1 occurring with event E_2 or its complement E_2^c . First, note that by definition of the complement

$$E_1 = (E_1 \cap E_2) \cup (E_1 \cap E_2^c) \quad (13.16)$$

Thus, since E_2 and E_2^c are a disjoint union, one has

$$\mathbb{P}(E_1) = \mathbb{P}(E_1 \cap E_2) + \mathbb{P}(E_1 \cap E_2^c) \quad (13.17)$$

or, in terms of conditional probabilities

$$\mathbb{P}(E_1) = \mathbb{P}(E_1|E_2)\mathbb{P}(E_2) + \mathbb{P}(E_1|E_2^c)\mathbb{P}(E_2^c) \quad (13.18)$$

This concept can be extended to any number of sets, E_2, \dots, E_n , which partition the entire outcome space into n disjoint events. This allows one to state the **law of total probability (LTP)** for event E_1 conditioned on all other events E_i for $i = 2, \dots, n$ as

$$\mathbb{P}(E_1) = \sum_{i=2}^n \mathbb{P}(E_1 \cap E_i) = \sum_{i=2}^n \mathbb{P}(E_1|E_i)\mathbb{P}(E_i) \quad (13.19)$$

The LTP also provides additional insight into Bayes theorem via the substitution for $\mathbb{P}(E_1)$ for two events, E_1 and E_2 , as

$$\mathbb{P}(E_2|E_1) = \frac{\mathbb{P}(E_1|E_2)\mathbb{P}(E_2)}{\mathbb{P}(E_1|E_2)\mathbb{P}(E_2) + \mathbb{P}(E_1|E_2^c)\mathbb{P}(E_2^c)} \quad (13.20)$$

or for more than two events, E_2, \dots, E_n , as

$$\mathbb{P}(E_i|E_1) = \frac{\mathbb{P}(E_1|E_i)\mathbb{P}(E_i)}{\sum_{i=2}^n \mathbb{P}(E_1|E_i)\mathbb{P}(E_i)} \quad i = 2, \dots, n \quad (13.21)$$

Note that Bayes' theorem allows one to calculate probabilities of events as more model-based information is obtained via sensing another related event, e.g., if one has modeled the prior probability for events E_2, \dots, E_n , then observes event E_1 , one can use that information to compute the posterior probability for events E_2, \dots, E_n , if one has a *probabilistic models* for their probabilities conditioned on E_1 .

Lastly, with these concepts in mind, one can define **independence** between events E_1 and E_2 as

$$\begin{aligned} \mathbb{P}(E_1) &= \mathbb{P}(E_1|E_2) = \mathbb{P}(E_1|E_2^c) \\ \mathbb{P}(E_2) &= \mathbb{P}(E_2|E_1) = \mathbb{P}(E_2|E_1^c) \end{aligned} \quad (13.22)$$

which implies the relationships

$$\begin{aligned}\mathbb{P}(E_1 \cap E_2) &= \mathbb{P}(E_1)\mathbb{P}(E_2) \\ \mathbb{P}(E_1 \cup E_2) &= \mathbb{P}(E_1) + \mathbb{P}(E_2) - \mathbb{P}(E_1)\mathbb{P}(E_2)\end{aligned}\tag{13.23}$$

It is important to note that independent is a property of events E_1, E_2 , and the measure \mathbb{P} while disjoint is a property only of the events E_1 and E_2 . Also, it can be shown that if E_1 and E_2 are independent, then E_1 and E_2^c , E_1^c and E_2 , and E_1^c and E_2^c are independent.

This concept can be extended to more than two events, $E_i, i = 1, \dots, n$, as **mutually independent** events if

$$\mathbb{P}\left(\bigcap_{i \in \mathcal{I}} E_i\right) = \prod_{i \in \mathcal{I}} \mathbb{P}(E_i)\tag{13.24}$$

for all possible finite subsets of events \mathcal{I} . However, if the previous equations holds only if $|\mathcal{I}|$ is restricted to 2, then events E_i are only **pairwise independent**.

Random Variables and Probability Distributions

A **random variable**, X , is a measurable function which returns values that *depend* on the outcomes of a random phenomenon, i.e., $X : \Omega \rightarrow \mathbb{S}$ where \mathbb{S} is called the **state space**, e.g., \mathbb{R} . This textbook will represent random variables by the uppercase letters, e.g. X , and the values or the **realizations**, by lowercase letters, e.g., x . Thus, $X = x$ denotes the random variable X is equal to the value x , which differentiates between the random variables and their particular realizations. A **discrete random variable** depends on outcomes that take on any value in a *countable* outcome set.

Thus, for discrete random variables, one can assign probabilities for individual outcomes, x_i , for all $i \in \mathbb{N}$. A **continuous random variable** depends on outcomes that take on any value in an uncountable outcome set. Thus, one cannot assign probabilities for the exact values of individual outcomes. Thus, though exact values still occur as part of the outcome space, one has the property $\mathbb{P}(X = x) = 0$. However, one can assign probabilities to *intervals* of values in the uncountable set as the events as mentioned in the previous section. These definitions can be shown to infer the following different probability distributions models for discrete and continuous random variables. Further details on the connections between random variables and axiomatic probability theory is beyond the scope of this textbook.

For modeling discrete random variables, one typically uses the **probability mass functions (PMFs)** defined for $x \in \mathbb{S} \subset \mathbb{R}$ as

$$p_X(x) = \mathbb{P}(X = x)\tag{13.25}$$

which must also satisfy that the probabilities of every possible event sum up to 100%, i.e.

$$\sum_i p_X(x_i) = \sum_{i=1}^{\infty} \mathbb{P}(X = x_i) = 1\tag{13.26}$$

thus, an event must always occur. A related function is the **probability-generating function (PGF)**

$$G_X(z) = \sum_{x=0}^{\infty} p_X(x)z^x\tag{13.27}$$

which exists for $x \in \mathbb{S} \subset \mathbb{R}$. One can see that the PGF is equivalent to the z -transform of the PMF. Here $p_X(x)$ can be obtained from the PGF by taking the derivatives of G_X via the formula

$$p_X(x) = \mathbb{P}(X = x) = \frac{G^{[x]}(0)}{x!} \quad (13.28)$$

For modeling continuous random variables, one typically uses **cumulative distribution functions (CDFs)** defined as

$$F_X(x) = \mathbb{P}(X \leq x) \quad (13.29)$$

From the properties of probability theory, the CDF is always a monotonically non-decreasing function, i.e., the accumulated probabilities as one moves from $-\infty$ to ∞ can never go down as an event cannot have a “negative” probability. Furthermore, the limits of the CDF must be

$$\lim_{x \rightarrow -\infty} F_X(x) = 0 \quad (13.30)$$

and

$$\lim_{x \rightarrow \infty} F_X(x) = 1 \quad (13.31)$$

thus, as one approaches $-\infty$, the probability that $X \leq -\infty$ should vanish and as one approaches ∞ , the probability that $X \leq \infty$ should approach 1. Furthermore, using the CDF, one can form the probability for X taking any realization between values a and b as

$$\mathbb{P}(a < X \leq b) = F_X(b) - F_X(a) \quad (13.32)$$

For absolutely continuous CDFs, one also typically considers the **probability density function (PDF)** of a continuous random variable defined as

$$f_X(x) = \frac{dF_X(x)}{dx} \quad (13.33)$$

or

$$F_X(x) = \int_{-\infty}^x f_X(\zeta) d\zeta \quad (13.34)$$

For $f_X(x)$ to be a properly defined PDF, the following equation must hold

$$\int_{-\infty}^{\infty} f_X(x) dx = 1 \quad (13.35)$$

It should be noted that one can intuitively think of the PDF as an infinitesimal PMF, i.e.

$$f_X(x) \approx \mathbb{P}(x < X \leq x + dx) \quad (13.36)$$

Two related functions are the **moment-generating function (MGF)**

$$M_X(s) = \int_{-\infty}^{\infty} f_X(x) e^{sx} dx \quad (13.37)$$

and the **characteristic function (CF)**

$$\varphi_X(\omega) = \int_{-\infty}^{\infty} f_X(x) e^{i\omega x} dx \quad (13.38)$$

where one can see that the MGF and CF are equivalent to the Laplace and Fourier transforms of the PDF, respectively. However, the MGF may not exist for all probability distributions while the CF always exists.

Lastly, the **complementary cumulative distribution function (CCDF)**, also known as the **tail distribution or exceedance**, is defined as

$$\bar{F}_X(x) = 1 - F_X(x) = \mathbb{P}(X \geq x) \quad (13.39)$$

Sometimes, this is combined into the **folded cumulative distribution function**

$$\tilde{F}_X(x) = F_X(x)1_{\{F_X(x) \leq 0.5\}} + \bar{F}_X(x)1_{\{F_X(x) > 0.5\}} \quad (13.40)$$

Depending on the properties of $F_X(x)$, the inverse of the CDF can be defined differently. If $F_X(x)$ strictly increasing and continuous, $F_X^{-1}(p)$ $p \in [0, 1]$ is the **inverse distribution function**, a.k.a. the **quantile function**. Otherwise, if there is no *unique* inverse, the inverse CDF can be defined as its **generalized inverse distribution function (GIDF)**, given by

$$F_X^{-1}(p) = \inf\{x \in \mathbb{R} : F_X(x) \geq p\} \quad \forall p \in [0, 1] \quad (13.41)$$

which preserves the inverse CDF properties

- F_X^{-1} : non-decreasing function
- $F_X^{-1}(xF(x)) \leq x$
- $F_X(F_X^{-1}(p)) \geq p$
- $F_X^{-1}(p) \leq x$ if and only if $p \leq F_X(x)$

One estimate of the CDF of a random variable is to take N independent samples. Then, one can state formally that if X_1, \dots, X_N are independent, identically distributed (IID) real random numbers with CDF, $F_X(x)$, then the **empirical distribution function (EDF)**, $\hat{F}_{X,N}(x)$, is given by the number of elements in sample $\leq x$ divided by number of samples, N , i.e.

$$\hat{F}_{X,N}(x) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{X_i \leq x}(x_i) \quad (13.42)$$

where $\mathbf{1}_A(x)$ is the **indicator** of event A , i.e.

$$\mathbf{1}_A(x) = \begin{cases} 1 & \text{if } x \in A \\ 0 & \text{if } x \notin A \end{cases} \quad (13.43)$$

One can quantify the statistical difference between the EDF and the CDF via the **Kolmogorov-Smirnov (KS) statistic** which is defined for a given CDF, $F_X(x)$, as

$$D_n = \sup_x |\hat{F}_{X,N}(x) - F_X(x)| \quad (13.44)$$

Intuitively, the KS statistic takes the largest absolute difference between the two distribution functions across all x values. By the **Gilvenko-Cantelli theorem** as $N \rightarrow \infty$, the EDF converges almost surely to the CDF, i.e.,

$$\hat{F}_{X,N}(x) \rightarrow F_X(x) \quad (13.45)$$

Lastly, the **Dvoretzky-Kiefer-Wolfowitz (DKW) inequality** uses the KS statistic to form a confidence interval for the CDF given an EDF through the inequality

$$\mathbb{P} \left(\sup_x |\hat{F}_{X,N}(x) - F_X(x)| > \epsilon \right) \leq 2 \exp(-2N\epsilon^2) \quad (13.46)$$

A **finite mixture distribution** is defined as a distribution that is a weighted sum of a finite N number of probability distributions, i.e., a discrete finite mixture distributions has a PMF as

$$p_X(x) = \sum_{i=1}^N w_i p_i(x) \quad (13.47)$$

and a continuous finite mixture distribution has a CDF as

$$F_X(x) = \sum_{i=1}^N w_i F_i(x) \quad (13.48)$$

where $p_i(x)$ is the i^{th} PMF term, $F_i(x)$ is the i^{th} CDF term, and $w_i \geq 0$ is the i^{th} weight with

$$\sum_{i=1}^N w_i = 1 \quad (13.49)$$

A **mixed random variable** allows some countable subset of outcomes to be assigned specific probabilities as part of an entire uncountable outcome space. These are formed as a mixture of both continuous and discrete probability distributions where the discrete values are modeled using the Dirac delta function, e.g., $F_i(x) = w_i \delta(x)$.

A **scale mixture random variable** is a random variable, Y , of the form

$$Y = \sqrt{Z} X \quad (13.50)$$

with $Z > 0$ independent from X . Here the distribution $X|(Z = z)$ is defined as the primary distribution and Z is the **scale parameter** or **generating variate**. Thus,

$$f_Y(x) = \int_0^\infty f_{X|Z}(x|z) f_Z(z) dz \quad (13.51)$$

is well-defined for a scale mixture.

Statistics of Random Variables

Furthermore, one often desires to compute some characteristic of the values a random variable may take, known as a **statistic**. There are many different types of statistics; however, many common statistics are defined using the expectation operator $\mathbb{E}[\bullet]$, which for the discrete random variable X is defined as

$$\mathbb{E}[X] = \sum_{i=1}^{\infty} x_i p_X(x_i) \quad (13.52)$$

and for continuous random variable X as

$$\mathbb{E}[X] = \int_{-\infty}^{\infty} x f_X(x) dx \quad (13.53)$$

which notably can also be computed for any function of a random variable. One of the most common functions are the **moments** of a random variable, where the i^{th} moment of a discrete random variable X is defined as

$$\mathbb{E}[X^i] = \sum_{j=1}^{\infty} x_j^i p_X(x_j) \quad (13.54)$$

and for continuous random variable X as

$$\mathbb{E}[X^i] = \int_{-\infty}^{\infty} x^i f_X(x) dx \quad (13.55)$$

It should be noted that this allows one to redefine the MGF for random variables as

$$\begin{aligned} M_X(s) &= \mathbb{E}[e^{sx}] \\ &= 1 + s\mathbb{E}[X] + \frac{s^2\mathbb{E}[X^2]}{2!} + \frac{s^3\mathbb{E}[X^3]}{3!} + \dots \end{aligned} \quad (13.56)$$

and for $Y = mX + b$ where m and b are scalars, one has

$$\begin{aligned} M_{mX+b}(s) &= \mathbb{E}[e^{s(mx+b)}] \\ M_{mX+b}(s) &= \mathbb{E}[e^{smx} e^{bs}] \\ M_Y(s) &= e^{bs} M_X(ms) \end{aligned} \quad (13.57)$$

In addition, the i^{th} **central moment** of a discrete random variable X is defined as

$$\mathbb{E}[(X - \mu_X)^i] = \sum_{j=1}^{\infty} (x_j - \mu_X)^i p_X(x_j) \quad (13.58)$$

and for continuous random variable X as

$$\mathbb{E}[(X - \mu_X)^i] = \int_{-\infty}^{\infty} (x - \mu_X)^i f_X(x) dx \quad (13.59)$$

The i^{th} **standardized moment** of a discrete random variable X is defined as

$$\mathbb{E} \left[\left(\frac{X - \mu_X}{\sigma_x} \right)^i \right] = \sum_{j=1}^{\infty} \left(\frac{x_j - \mu_X}{\sigma_x} \right)^i p_X(x_j) \quad (13.60)$$

and for continuous random variable X as

$$\mathbb{E} \left[\left(\frac{X - \mu_X}{\sigma_x} \right)^i \right] = \int_{-\infty}^{\infty} \left(\frac{x - \mu_X}{\sigma_x} \right)^i f_X(x) dx \quad (13.61)$$

Four moments of random variables have particular names. These are the **mean**, denoted by μ_X , which is the first moment, the **variance**, denoted by σ_x^2 , which is second central moment, the **skewness** which is the third standardized moment, and the **kurtosis** which is the fourth standardized moment. Three other common statistics of a random variable X are the **median**, denoted by m_x , and defined by the equation

$$m_X = F_X^{-1}(0.5) \quad (13.62)$$

the **mode**, denoted by Mo_x , and defined by

$$Mo_X = \operatorname{argmax}_x f_X(x) \quad (13.63)$$

and the **standard deviation**, denoted by σ_X , and defined by

$$\sigma_X = \sqrt{\sigma_x^2} \quad (13.64)$$

References

For more information, please refer to the following

- Gubner, J.A., “Chapter 1 Introduction to Probability,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 1-47
- Gubner, J.A., “Chapter 2 Introduction to Discrete Random Variables,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 1-47
- Gubner, J.A., “Chapter 3 More About Discrete Random Variables,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 1-47
- Gubner, J.A., “Chapter 4 Continuous Random Variables,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 1-47
- Gubner, J.A., “Chapter 5 Cumulative Distributions Functions and Their Applications,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 1-47

- Simon, D., “2.1 Probability,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 49-53
- Simon, D., “2.2 Random Variables,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 53-59
- Simon, D., “2.3 Transformations of Random Variables,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 59-61

13.2 Random Vectors

When considering a finite collection of n random variables, one has a **random vector** with dimension n defined as

$$\vec{X} = \begin{bmatrix} X_1 \\ \vdots \\ X_n \end{bmatrix} \quad (13.65)$$

with a corresponding **realization vector** that can be defined as

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \quad (13.66)$$

Probability Functions of Random Vectors

To model the probabilities and relationships between the random elements of a discrete random vector, one typically uses the **joint probability mass function (joint PMF)**, defined as

$$p_{X_1, \dots, X_n}(x_1, \dots, x_n) = \mathbb{P}(X_1 = x_1, \dots, X_n = x_n) \quad (13.67)$$

which can be written in vector form as

$$p_{\vec{X}}(\vec{x}) = \mathbb{P}(\vec{X} = \vec{x}) \quad (13.68)$$

To model the probabilities and relationships between the random elements of a continuous random vector, one typically uses the **joint cumulative distribution function (joint CDF)**, defined as

$$F_{X_1, \dots, X_n}(x_1, \dots, x_n) = \mathbb{P}(X_1 \leq x_1, \dots, X_n \leq x_n) \quad (13.69)$$

which can be written in vector form as

$$F_{\vec{X}}(\vec{x}) = \mathbb{P}(\vec{X} \leq \vec{x}) \quad (13.70)$$

Furthermore, by taking the n partial derivatives of the joint CDF, one can define the **joint probability density function (joint PDF)** as

$$f_{\vec{X}}(\vec{x}) = \frac{\partial^n F_{\vec{X}}(\vec{x})}{\partial x_1 \cdots \partial x_n} \quad (13.71)$$

which can be considered intuitively as

$$\mathbb{P}(x_1 < X_1 \leq x_1 + dx_1, \dots, x_n < X_n \leq x_n + dx_n) \quad (13.72)$$

Reversing this relationship, one can also relate the joint CDF and joint PDF by

$$F_{\vec{X}}(\vec{x}) = \int_{-\infty}^{x_1} \cdots \int_{-\infty}^{x_n} f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (13.73)$$

where for any PDF, including a joint PDF, the following is true.

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n = 1 \quad (13.74)$$

Similarly to random variables, three related functions are the **probability-generating function (PGF)**

$$G_{\vec{X}}(\vec{z}) = \sum_{\vec{x}=0}^{\infty} p_{\vec{X}}(\vec{x}) z_1^{x_1} \cdots z_n^{x_n} \quad (13.75)$$

the **moment-generating function (MGF)**

$$M_{\vec{X}}(\vec{s}) = \mathbb{E}[\exp \vec{s}^T \vec{x}] \quad (13.76)$$

and the **characteristic function (CF)**

$$\varphi_{\vec{X}}(\vec{\omega}) = \mathbb{E}[\exp i \vec{\omega}^T \vec{x}] \quad (13.77)$$

The **empirical distribution function (EDF)** for random vectors, $\hat{F}_{\vec{X},N}(\vec{x})$, can be defined analogously to the random variable case, i.e.,

$$\hat{F}_{\vec{X},N}(\vec{x}) = \frac{1}{N} \sum_{i=1}^N \mathbf{1}_{X_i \leq x}(x_i) \quad (13.78)$$

and the multivariate **Dvoretzky-Kiefer-Wolfowitz (DKW) inequality** uses the KS statistic to form a confidence interval for the joint CDF given an EDF through the inequality

$$\mathbb{P} \left(\sup_x |\hat{F}_{\vec{X},N}(\vec{x}) - F_{\vec{X}}(\vec{x})| > \epsilon \right) \leq 2n_x \exp(-2N\epsilon^2) \quad (13.79)$$

Bayes' Rule for Random Vectors

For discrete random vectors, the **conditional PMF** of \vec{X} as the probability of $\vec{X} = \vec{x}$ given $\vec{Y} = \vec{y}$, i.e.,

$$p_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = \frac{p_{\vec{X},\vec{Y}}(\vec{x}, \vec{y})}{p_{\vec{Y}}(\vec{y})} \quad (13.80)$$

Extending Bayes' rule for discrete random variables, \vec{X} and \vec{Y} , one has

$$p_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = \frac{p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x}) p_{\vec{X}}(\vec{x})}{p_{\vec{Y}}(\vec{y})} \quad (13.81)$$

Extending the **law of total probability (LTP)** for discrete random variables, one can relate marginal and conditional PMFs as

$$p_{\vec{Y}}(\vec{y}) = \sum_{\vec{x}} p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})p_{\vec{X}}(\vec{x}) \quad (13.82)$$

Note that by using the LTP for $f_{\vec{Y}}(\vec{y})$, one also has for Bayes' Rule

$$p_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = \frac{p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})p_{\vec{X}}(\vec{x})}{\sum_{\vec{x}} p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})p_{\vec{X}}(\vec{x})} \quad (13.83)$$

where the denominator essentially serves as a normalizing factor for the product $p_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})p_{\vec{X}}(\vec{x})$ as $\sum_{\vec{x}} p_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = 1$ by definition of a PMF.

For continuous random vectors, the **marginal PDF** can be defined for any element of \vec{X} as

$$f_{X_i}(x_i) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{X}}(\vec{x}) dx_1 \cdots dx_{i-1} dx_{i+1} \cdots dx_n \quad (13.84)$$

which can also be generalized to be any sub-vector. Thus, the general marginal PDF also leads to the random vector definition for the **conditional PDF** of \vec{X} as the probability of \vec{X} given $\vec{Y} = \vec{y}$, i.e.

$$f_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = \frac{f_{\vec{X},\vec{Y}}(\vec{x},\vec{y})}{f_{\vec{Y}}(\vec{y})} \quad (13.85)$$

Extending Bayes' rule for continuous random variables, \vec{X} and \vec{Y} , one has

$$f_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = \frac{f_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})f_{\vec{X}}(\vec{x})}{f_{\vec{Y}}(\vec{y})} \quad (13.86)$$

Extending the **law of total probability (LTP)** for continuous random variables, one can relate marginal and conditional PDFs as

$$f_{\vec{Y}}(\vec{y}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{Y}|\vec{X}}(\vec{y}|\vec{\zeta})f_{\vec{X}}(\vec{\zeta})d\vec{\zeta} \quad (13.87)$$

Note that by using the LTP for $f_{\vec{Y}}(\vec{y})$, one also has for Bayes' Rule

$$f_{\vec{X}|\vec{Y}}(\vec{x}|\vec{y}) = \frac{f_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})f_{\vec{X}}(\vec{x})}{\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{Y}|\vec{X}}(\vec{y}|\vec{\zeta})f_{\vec{X}}(\vec{\zeta})d\vec{\zeta}} \quad (13.88)$$

where the denominator essentially serves as a normalizing factor for the product $f_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})f_{\vec{X}}(\vec{x})$ as $\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{Y}|\vec{X}}(\vec{y}|\vec{x})f_{\vec{X}}(\vec{x})dx = 1$ by definition of a PDF.

Functions of Random Vectors

In perception systems, one often uses functions of random vectors, in particular, invertible functions, h , of random vectors, e.g.

$$\vec{Y} = h(\vec{X}) \quad (13.89)$$

For discrete random variables, the PMF of \vec{Y} in terms of the PMF of \vec{X} is

$$p_{\vec{Y}}(\vec{y}) = \mathbb{P}(\vec{Y} = \vec{y}) = \mathbb{P}(h(\vec{X}) = \vec{y}) = \mathbb{P}(\vec{X} = h^{-1}(\vec{y})) = p_{\vec{X}}(h^{-1}(\vec{y})) \quad (13.90)$$

and for continuous random variables, the CDF of \vec{Y} in terms of the CDF of \vec{X} is

$$F_{\vec{Y}}(\vec{y}) = \mathbb{P}(\vec{Y} \leq \vec{y}) = \mathbb{P}(h(\vec{X}) \leq \vec{y}) = \begin{cases} \mathbb{P}(\vec{X} \leq h^{-1}(\vec{y})) = F_{\vec{X}}(h^{-1}(\vec{y})) & h^{-1} \text{ increasing} \\ \mathbb{P}(\vec{X} \geq h^{-1}(\vec{y})) = 1 - F_{\vec{X}}(h^{-1}(\vec{y})) & h^{-1} \text{ decreasing} \end{cases} \quad (13.91)$$

which by the definition of the CDF and derivatives, one can show that the PDF of \vec{Y} given \vec{X} is

$$f_{\vec{Y}}(\vec{y}) = \left[\frac{f_{\vec{X}}(\vec{x})}{\left| \det \frac{\partial h(\vec{x})}{\partial \vec{x}} \right|} \right]_{\vec{x}=h^{-1}(\vec{y})} \quad (13.92)$$

which is generally difficult to handle directly for general functions, but is approximated using many different methods.

However, a closed-form solution exists if $h()$ is the **affine transformation**

$$\vec{Y} = h(\vec{x}) = H\vec{X} + \vec{b} \quad (13.93)$$

Thus,

$$h^{-1}(\vec{y}) = H^{-1}(\vec{y} - \vec{b}) \quad (13.94)$$

and

$$\frac{\partial h(\vec{x})}{\partial \vec{x}} = H \quad (13.95)$$

then, the PDF of \vec{Y} can be written as

$$f_{\vec{Y}}(\vec{y}) = \frac{f_{\vec{X}}(H^{-1}(\vec{y} - \vec{b}))}{| \det H |} \quad (13.96)$$

Another important result for perception is that the PMF of the sum of two independent discrete random vectors $\vec{Z} = \vec{X} + \vec{Y}$ is given by

$$p_{\vec{Z}}(\vec{z}) = \mathbb{P}(\vec{Z} = \vec{z}) = \sum_{\vec{x}} \mathbb{P}(\vec{Z} = \vec{z} | \vec{X} = \vec{x}) \mathbb{P}(\vec{X} = \vec{x}) \quad (13.97)$$

which by substitution for the event that $\vec{Z} = \vec{z} | \vec{X} = \vec{x}$ as $\vec{Y} = \vec{z} - \vec{x}$ since $\vec{Y} = \vec{Z} - \vec{X}$ is

$$p_{\vec{Z}}(\vec{z}) = \sum_{\vec{x}} \mathbb{P}(\vec{Y} = \vec{z} - \vec{x}) \mathbb{P}(\vec{X} = \vec{x}) \quad (13.98)$$

for independent discrete random variables \vec{X} and \vec{Y} , one has

$$p_{\vec{Z}}(\vec{z}) = \sum_{\vec{x}} p_{\vec{Y}}(\vec{z} - \vec{x}) p_{\vec{X}}(\vec{x}) \quad (13.99)$$

which is a convolution summation between the PMFs of \vec{X} and \vec{Y} . Thus, due to the properties of the z-transform, this can be written in terms of the PGFs as

$$G_{\vec{Z}}(\vec{z}) = G_{\vec{X}}(\vec{z})G_{\vec{Y}}(\vec{z}) \quad (13.100)$$

Likewise, the sum of two independent continuous random variables $\vec{Z} = \vec{X} + \vec{Y}$ is given by

$$F_{\vec{Z}}(\vec{z}) = \mathbb{P}(\vec{Z} \leq \vec{z}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{Z}|\vec{X}}(\vec{z}|\vec{x})f_{\vec{X}}(\vec{x})d\vec{x} \quad (13.101)$$

which by substitution for the event that $\vec{Z} \leq \vec{z} | \vec{X} = \vec{x}$ as $\vec{Y} \leq \vec{z} - \vec{x}$ since $\vec{Y} = \vec{Z} - \vec{X}$ is

$$f_{\vec{Z}}(\vec{z}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{Y}}(\vec{z} - \vec{x})f_{\vec{X}}(\vec{x})d\vec{x} \quad (13.102)$$

which is a convolution integral between the PDFs of independent discrete random variables \vec{X} and \vec{Y} . Thus, due to the properties of the Fourier transform, this can be written in terms of the CFs as

$$\varphi_{\vec{Z}}(\vec{z}) = \varphi_{\vec{X}}(\vec{z})\varphi_{\vec{Y}}(\vec{z}) \quad (13.103)$$

Statistics of Random Vectors

To characterize the statistics of different random vectors, one can use the expectation operator which operates on each individual random variable within the vector, i.e.

$$\mathbb{E}[\vec{X}] = \begin{bmatrix} \mathbb{E}[X_1] \\ \vdots \\ \mathbb{E}[X_n] \end{bmatrix} \quad (13.104)$$

For discrete random vectors

$$\mathbb{E}[\vec{X}] = \sum_{\vec{x}} \vec{x} p_{\vec{X}}(\vec{x}) \quad (13.105)$$

and for continuous random vectors, one has

$$\mathbb{E}[\vec{X}] = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \vec{x} f_{\vec{X}}(\vec{x})d\vec{x} \quad (13.106)$$

Thus, the multivariate analog for the first moment of \vec{X} is the **mean of a random vector** is simply

$$\mu_{\vec{X}} = \begin{bmatrix} \mu_{X_1} \\ \vdots \\ \mu_{X_n} \end{bmatrix} \quad (13.107)$$

However, higher moments of random vectors, one must consider the relationship *between* the random variables within the vector. Thus, the multivariate analog of the second moment of \vec{X} is the **correlation matrix** defined as

$$R_{XX} = \mathbb{E}[\vec{X}\vec{X}^T] = \begin{bmatrix} \mathbb{E}[X_1^2] & \cdots & \mathbb{E}[X_1 X_n] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[X_n X_1] & \cdots & \mathbb{E}[X_n^2] \end{bmatrix} \quad (13.108)$$

while the second centralized moment of \vec{X} is the **covariance matrix**, also known as the **variance-covariance matrix**, defined as

$$\begin{aligned} C_{XX} &= \mathbb{E}[(\vec{X} - \bar{x})(\vec{X} - \mu_{\vec{X}})^T] = \begin{bmatrix} \mathbb{E}[(X_1 - \mu_{X_1})^2] & \cdots & \mathbb{E}[(X_1 - \mu_{X_1})(X_n - \mu_{X_n})] \\ \vdots & \ddots & \vdots \\ \mathbb{E}[(X_n - \mu_{X_n})(X_1 - \mu_{X_1})] & \cdots & \mathbb{E}[(X_n - \mu_{X_n})^2] \end{bmatrix} \\ &= \begin{bmatrix} \sigma_1^2 & \cdots & \sigma_{n,1} \\ \vdots & \ddots & \vdots \\ \sigma_{1,n} & \cdots & \sigma_n^2 \end{bmatrix} \end{aligned} \quad (13.109)$$

which is a symmetric positive definite matrix. It can easily be shown that

$$R_{XX} = C_{XX} + \mu_{\vec{X}}\mu_{\vec{X}}^T \quad (13.110)$$

The individual elements of the covariance matrix are called the variances on the diagonal, i.e.

$$\sigma_i^2 = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (x_i - \mu_{X_i})^2 f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (13.111)$$

and the covariances on the off-diagonal, i.e.

$$\sigma_{i,j} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} (x_i - \mu_{X_i})(x_j - \mu_{X_j}) f_{\vec{X}}(x_1, \dots, x_n) dx_1 \dots dx_n \quad (13.112)$$

Lastly, the inverse of the covariance is known as the **precision matrix**.

Furthermore, if one relates the correlation for each covariance element in C_{XX} , one has

$$\sigma_{i,j} = \rho_{i,j} \sigma_i \sigma_j \quad (13.113)$$

where $\rho_{i,j} \in [-1, 1]$ is called the **correlation coefficient** between X_i and X_j . Thus, the elements of \vec{X} are **uncorrelated** if $\rho_{i,j} = \sigma_{i,j} = 0$, or in terms of expectations

$$\mathbb{E}[X_i X_j] = \mathbb{E}[X_i] \mathbb{E}[X_j] \quad \forall i \neq j \quad (13.114)$$

If the elements of \vec{X} are uncorrelated for all $i \neq j \rightarrow$, then the correlation matrix is a diagonal matrix. Furthermore, the elements of \vec{X} are **pairwise independent** if

$$p_{X_i, X_j}(x_i, x_j) = p_{X_i}(x_i)p_{X_j}(x_j) \quad \text{or} \quad F_{X_i, X_j}(x_i, x_j) = F_{X_i}(x_i)F_{X_j}(x_j) \quad \forall i \neq j \quad (13.115)$$

It can be shown that independence implies uncorrelation, however, it does not hold vice versa.

It should be noted that this concepts of covariance, correlation, and independence can be extended to *different* random vectors, e.g. \vec{X} and \vec{Y} . The correlation matrix can be generalized to the **cross-correlation matrix** defined as

$$R_{XY} = \mathbb{E}[\vec{X} \vec{Y}^T] \quad (13.116)$$

The covariance matrix can be generalized to the **cross-covariance matrix** defined as

$$C_{XY} = \mathbb{E} [(\vec{X} - \mu_{\vec{X}})(\vec{Y} - \mu_{\vec{Y}})^T] = R_{XY} - \mu_{\vec{X}}\mu_{\vec{Y}}^T \quad (13.117)$$

Independence between \vec{X} and \vec{Y} occurs if

$$p_{\vec{X}, \vec{Y}}(\vec{x}, \vec{y}) = p_{\vec{X}}(\vec{x})p_{\vec{Y}}(\vec{y}) \quad \text{or} \quad F_{\vec{X}, \vec{Y}}(\vec{x}, \vec{y}) = F_{\vec{X}}(\vec{x})F_{\vec{Y}}(\vec{y}) \quad \forall \vec{x}, \vec{y} \quad (13.118)$$

Statistical distances, denoted as $D(X, Y)$, are used to determine the degree of similarity between two statistical objects X and Y , i.e., between realizations or between probability distributions. Distances are important to perception system design for determining an algorithm's performance, e.g., the error of estimation with respect to some truth source, and for comparing different algorithms with each other. A statistical distance, $D(X, Y)$ is called a **metric** if it satisfies the following conditions:

- 1 Non-negativity: $D(X, Y) \geq 0$
- 2 Identity of indiscernibles: $D(X, Y) = 0$ if and only if $X = Y$
- 3 Symmetry: $D(X, Y) = D(Y, X)$
- 4 Triangle inequality: $D(X, Z) \leq D(X, Y) + D(Y, Z)$

where a **pseudometric** does not satisfy 2, a **quasimetric** does not satisfy 3, a **semimetric** does not satisfy 4, and a **divergence** satisfies 1 and 2.

One common metric for perception systems is the **Mahalanobis distance** which can be defined between the probability distribution, X on \mathbb{R}^n , with mean $\vec{\mu}_x$ and covariance P , and the probability distribution, Y on \mathbb{R}^n , with mean $\vec{\mu}_y$ and the same covariance, P , as

$$D_M(X, Y) = \|\vec{\mu}_x - \vec{\mu}_y\|_{P^{-1}} = \sqrt{(\vec{\mu}_x - \vec{\mu}_y)P^{-1}(\vec{\mu}_x - \vec{\mu}_y)} \quad (13.119)$$

or, alternatively, between the realization $\vec{x} \in \mathbb{R}^n$ and the probability distribution, Y , as

$$D_M(\vec{x}, Y) = \|\vec{x} - \vec{\mu}_y\|_{P^{-1}} = \sqrt{(\vec{x} - \vec{\mu}_y)P^{-1}(\vec{x} - \vec{\mu}_y)} \quad (13.120)$$

One common divergence for perception systems is the **Kullback-Leibler (KL) divergence**, also known as the **relative entropy** or **information gain**, which can be defined for the discrete probability distribution, X on \mathcal{X} , with PMF $p_{\vec{X}}(\vec{x})$, and the discrete probability distribution, Y on \mathcal{X} , with PMF $p_{\vec{Y}}(\vec{y})$, as

$$D_{KL}(X||Y) = \sum_{\vec{x} \in \mathcal{X}} p_{\vec{X}}(\vec{x}) \log \left(\frac{p_{\vec{X}}(\vec{x})}{p_{\vec{Y}}(\vec{x})} \right) \quad (13.121)$$

or, alternatively, between the continuous probability distribution, X on \mathbb{R}^n , with PDF $f_{\vec{X}}(\vec{x})$, and the continuous probability distribution, Y on \mathbb{R}^n , with PDF $f_{\vec{Y}}(\vec{x})$, as

$$D_{KL}(X||Y) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{X}}(\vec{x}) \log \left(\frac{f_{\vec{X}}(\vec{x})}{f_{\vec{Y}}(\vec{x})} \right) d\vec{x} \quad (13.122)$$

This is typically referred to as the “divergence of X from Y ” or the “information gain from X over Y .” In a Bayesian context, Y is the prior distribution and X is the posterior distribution.

Samples of Random Vectors

A sample realization of a random variable or vector is known as a **random variate**. Thus, generating samples of random variables is known as **random variate generation** which can be uniform or non-uniform, also known as true **random number generation (RNG)** or **pseudo-random number generator (PRNG)**, respectively. Both of these types of generation are used in control and perception system modeling and simulations. True RNGs depend on some natural phenomenon that can be sampled using hardware, e.g., thermal noise, shot noise, jitter, and metastability of electronic circuits. However, in many applications, one can approximate the statistical nature of RNG using PRNG algorithms.

The basic method of PRNG utilizes the **uniform distribution**, represented by $U \sim \mathcal{U}(a, b)$, which has a CDF of the form

$$F_X(u; a, b) = \begin{cases} 0 & u < a \\ \frac{u-a}{b-a} & u \in [a, b] \\ 1 & u > b \text{ otherwise} \end{cases} \quad (13.123)$$

and PDF of the form

$$f_X(u; a, b) = \begin{cases} \frac{1}{b-a} & u \in [a, b] \\ 0 & \text{otherwise} \end{cases} \quad (13.124)$$

and can be used to model events when all values in the range $[a, b]$ are equally likely. Then, for generating a random variate, x , from an arbitrary probability distribution with CDF, $F_X(x)$, one can use **inverse transform sampling**, also known as the **Smirnov transform**, by generating a random variate, u , from $U \sim \mathcal{U}(0, 1)$ and computing

$$x = F_X^{-1}(u) \quad (13.125)$$

where F_X^{-1} is the generalized inverse CDF.

Notably, the inverse transform sampling requires an analytical expression or numerical tables for the CDF. Thus, an alternative sampling method is **rejection sampling**, also known as **acceptance-rejection sampling**, where one generates a random variate, y , from **proposal distribution** with PDF, f_Y , and a random variate, u , from $U \sim \mathcal{U}(0, 1)$ and “accept” y as drawn from X with PDF, f_X , if

$$u < \frac{f_X(y)}{M f_Y(y)} \quad (13.126)$$

where $1 \geq M < \infty$ is chosen to satisfy $f_X(y) \leq M f_Y(y)$ for all values of x and a sample will be “accepted” after M iterations on average. Notably, the support of Y must include the support of X . In practice, a lower value of M is preferred as there will be fewer rejected samples; however, this requires that $f_Y(y)$ should resemble $f_X(y)$ in some fashion while also satisfying its bounding property.

For sampling random vectors of dimension n , one can take advantage of **Sklar's theorem** which states that every joint CDF can be decomposed into its marginal CDFs and a **copula**, i.e.,

$$F_{\vec{X}}(\vec{x}) = C(F_{X_1}(x_1), \dots, F_{X_n}(x_n)) \quad (13.127)$$

where $C : [0, 1]^n \rightarrow [0, 1]$ is an n -dimensional copula which is a specific type of joint CDF. Furthermore, if the joint PDF is available, one also has

$$f_{\vec{X}}(\vec{x}) = c(f_{X_1}(x_1), \dots, f_{X_n}(x_n)) \quad (13.128)$$

where c is the density of the n -dimensional copula. Thus, by sampling each marginal distribution via inverse transform or rejection sampling, one can simply compute the output of the joint CDF via the copula. Intuitively, one can think of the copula as directly modeling the dependence between random variables. Often, copulas are used in estimating the distributions of random vectors by separating the marginal estimation problems and the copula estimation problem.

References

For more information, please refer to the following

- Gubner, J.A., “Chapter 6 Introduction to Discrete Random Variables,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 1-47
- Naaman, M., “On the Tight Constant in the Multivariate Dvoretzky–Kiefer–Wolfowitz Inequality,” in *Statistics and Probability Letters*, Vol. 173, 2021, pp. 1–8.
- Simon, D., “2.4 Multiple Random Variables,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 61-67
-

13.3 Fundamental Probability Distributions for Perception

Fundamental Discrete Probability Distributions

The **Poisson distribution**, denoted by $X \sim Pois(\lambda)$, has a PMF of the form

$$p_X(x; \lambda) = \frac{\lambda^x e^{-\lambda}}{x!} \quad (13.129)$$

with $\lambda > 0$ is the Poisson rate and can be used to model the number of events, x , that may occur in some time interval. The mean and variance of the Poisson distribution is λ .

The **Bernoulli distribution**, denoted by $X \sim Ber(r)$, has a PMF of the form

$$p_X(x; r) = \begin{cases} 1 - r & \text{if } X = x_1 \\ r & \text{if } X = x_2 \end{cases} \quad (13.130)$$

and can be used to model event with only two outcomes, e.g., existence or non-existence.

The **multi-Bernoulli distribution**, denoted by $X \sim MBer(r_1, \dots, r_N)$, is given by the union of N independent Bernoulli-distributed random variables and has a PMF of the form

$$p_X(x; N, p) = \begin{cases} \prod_{j=1}^N (1 - r_j) & \text{if } X_1 = \dots = X_N = x_1 \\ \vdots & \vdots \\ \prod_{j=1}^N (1 - r_j) \sum_{1 \leq i_1 \neq \dots \neq i_n \leq N} \prod_{j=1}^n \frac{r_{i_j}}{1 - r_{i_j}} & \text{if } X_{i_1} = \dots = X_{i_n} = x_1, X_{i_{n+1}} = \dots = X_{i_N} = x_2 \\ \vdots & \vdots \\ \prod_{j=1}^N r_{i_j} & \text{if } X_1 = \dots = X_N = x_2 \end{cases} \quad (13.131)$$

where the sum for each intermediate outcome is taken over all permutations of $n \leq N$ of the N constituent Bernoulli-distributed random variables. The numerator is the probability density that the Bernoulli components with indices i_1, \dots, i_n generate realizations x_2 . The leading constant cancels with the divisors inside the sum to give the probability that the leftover Bernoulli components with indices i_{n+1}, \dots, i_N generate realizations x_1 . This can be used to model N simultaneous events with only two outcomes, e.g., N existences or non-existences.

Fundamental Continuous Univariate Probability Distributions

The **Gaussian distribution**, also known as the **normal distribution**, denoted by $X \sim \mathcal{N}(\mu, \sigma^2)$, has a PDF of the form

$$f_X(x; \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{1}{2}\frac{(x-\mu)^2}{\sigma^2}\right] \quad (13.132)$$

where μ is the mean and σ^2 is the variance. The **standard normal distribution** has $\mu = 0$ and $\sigma^2 = 1$ which provides the simpler PDF

$$f_Y(y) = \frac{1}{\sqrt{2\pi}} \exp\left[-\frac{1}{2}y^2\right] \quad (13.133)$$

and thereby rewrite $f_X(x; \mu, \sigma^2)$ in terms of $f_Y(y)$ as

$$f_X(x; \mu, \sigma^2) = \frac{1}{\sigma} f_Y\left(\frac{x-\mu}{\sigma}\right) \quad (13.134)$$

It should be noted that while an analytical form exists for the PDF of a Gaussian, the CDF doesn't not admit an analytical solution and must be numerically approximated.

The mean of a Gaussian distribution is also the median and mode. The skewness of a Gaussian is 0 and the kurtosis is 3. The prevalence of the Gaussian distribution leads to the definition of the **excess kurtosis** as the kurtosis minus 3. The Gaussian distribution typically is used for computing **confidence regions** as the probability that a realization x of $X \sim \mathcal{N}(\mu, \sigma^2)$ lies within some interval around the mean. Typically values are

$$\Pr(\mu - \sigma < X \leq \mu + \sigma) = 0.68 \quad (13.135)$$

$$\Pr(\mu - 2\sigma < X \leq \mu + 2\sigma) = 0.95 \quad (13.136)$$

$$\Pr(\mu - 3\sigma < X \leq \mu + 3\sigma) = 0.997 \quad (13.137)$$

The **Cauchy distribution**, denoted by $X \sim \text{Cauchy}(x_0, \gamma)$, has a PDF of the form

$$f_X(x; x_0, \gamma) = \left(\pi \gamma \left[1 + \left(\frac{x - x_0}{\gamma} \right)^2 \right] \right)^{-1} \quad (13.138)$$

where x_0 is the location and γ is the scale. Notably, the mean, variance, skewness, and kurtosis for the Cauchy are undefined. However, x_0 is the median and the mode. The **standard Cauchy distribution** occurs for $x_0 = 0$ and $\gamma = 1$. Notably, if $X \sim \mathcal{N}(0, 1)$ and $Y \sim \mathcal{N}(0, 1)$, then $X/Y \sim \text{Cauchy}(0, 1)$

The **Pareto distribution**, denoted by $X \sim \text{Pareto}(\gamma, \beta)$, has a PDF of the form

$$f_X(x; \gamma, \beta) = \begin{cases} \frac{1}{\beta} \left(1 + \frac{\gamma}{\beta} x \right)^{-\frac{\gamma+1}{\gamma}} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (13.139)$$

where $\gamma \geq 0$ is the shape and $\beta > 0$ is the scale. For $\gamma = 0$, one has

$$f_X(x; 0, \beta) = \begin{cases} e^{-\frac{x}{\beta}} & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (13.140)$$

The Pareto distribution is used to model heavy-tailed distribution, F_X , due to its appearance in **Extreme Value Theory (EVT)** as a limiting distribution, as the threshold $u \rightarrow \infty$, of its conditional excess distribution function (CEDF) defined as

$$F_u(y) = \mathbb{P}(X - u \leq y | X > u) = \frac{F_X(u + y) - F_X(u)}{1 - F_X(u)} \quad (13.141)$$

A **heavy-tailed distribution** is defined as a probability distributions whose tails are not exponentially bounded, i.e.,

$$\lim_{x \rightarrow \infty} e^{tx} \bar{F}_X(x) = \infty \quad \forall t > 0 \quad (13.142)$$

The **Levy α -stable (α S) distribution**, denoted by $X \sim S(\alpha, \beta, c, \mu)$, has a CF of the form

$$\varphi_{\vec{X}}(\omega; \alpha, \beta, c, \mu) = \exp(i\mu\omega - |c\omega|^\alpha(1 - i\beta\text{sign}(\omega)\Phi)) \quad (13.143)$$

where

$$\Phi = \begin{cases} \tan\left(\frac{\pi\alpha}{2}\right) & \alpha \neq 1 \\ -\frac{2}{\pi} \log|\omega| & \alpha = 1 \end{cases} \quad (13.144)$$

where $0 < \alpha \leq 2$ is the **stability index**, also known as the **tail index** or **characteristic exponent**, $-1 \leq \beta \leq 1$ is the skewness, $c > 0$ is the scale parameter, and $\mu \in \mathbb{R}$ is the shift parameter. The **symmetric α -stable ($S\alpha$ S) distribution** occurs for $\beta = 0$. Notably, the Gaussian distribution occurs for $\alpha = 2$ and $\beta = 0$, the **Cauchy distribution** occurs for $\alpha = 1$ and $\beta = 0$, and the **Levy distribution** occurs for $\alpha = 0.5$ and $\beta = 1$. Other than these three, no analytical form exists for the CDF or PDF.

Furthermore, the variance, skewness, and kurtosis are only defined for $\alpha = 2$ and the mean is not defined for $\alpha \leq 1$. The reason for the name is that α -stable distributions are closed under convolution for a fixed value of α , i.e., sums of independent α -stable distributions remain α -stable if α is constant, albeit β , γ , and δ may change.

An important theorem in perception is the **generalized central limit theorem** which states the sum of IID random variables with distributions having infinite variance, i.e., power-law “Pareto” tails, decreasing as $|x|^{-\alpha-1}$ where $0 < \alpha \leq 2$, will tend to an α -stable distribution $S(\alpha, 0, c, 0)$ as the number of summands grows. If the symmetric distributions have finite variance, i.e., $\alpha > 2$, then the **central limit theorem** states the sum converges to a Gaussian distribution, i.e., a stable distribution with stability parameter equal to 2.

The (**student's**) **t distribution**, denoted by $X \sim t(\nu)$, has a PDF of the form

$$f_X(x; \nu) = \frac{\Gamma(\frac{\nu+1}{2})}{\sqrt{\nu\pi}\Gamma(\frac{\nu}{2})} \left(1 + \frac{x^2}{\nu}\right)^{-\frac{\nu+1}{2}} \quad (13.145)$$

where $\nu > 0$ is the degrees of freedom and Γ is the gamma function

$$\Gamma(n) = (n - 1)! \quad (13.146)$$

Notably, the **standard Cauchy distribution** occurs for $\nu = 1$ and the standard normal distribution occurs as $\nu \rightarrow \infty$. Thus, the t distribution can be used to model heavier tails than the standard normal distribution. The mean of the t distribution is 0 for $\nu > 1$ and undefined otherwise. The variance of the t distribution is $\nu(\nu - 2)^{-1}$ for $\nu > 1$ and undefined otherwise. The skewness of the t distribution is 0 for $\nu > 3$ and undefined otherwise. The excess kurtosis of the t distribution is $6(\mu - 4)^{-1}$ for $\nu > 4$, ∞ for $2 < \nu \leq 4$, and undefined otherwise.

The **Γ distribution**, denoted by $X \sim \Gamma(\alpha, \theta)$, has a PDF of the form

$$f_X(x; \alpha, \beta) = \frac{x^{\alpha-1} e^{-\beta x} \beta^\alpha}{\Gamma(\alpha)} \quad (13.147)$$

where $\alpha > 0$ is the shape parameter and $\beta > 0$ is the rate parameter. Two special cases of the Γ distribution are used in perception systems. The **exponential distribution**, denoted by $X \sim Exp(\lambda)$, occurs when $X \sim \Gamma(1, \lambda)$ has a PDF of the form

$$f_X(x; \beta) = \begin{cases} \lambda \exp(-\lambda x) & x \geq 0 \\ 0 & x < 0 \end{cases} \quad (13.148)$$

where $\lambda > 0$ is the rate parameter. The mean of the exponential is β and the variance is β^2 . The **central χ^2 distribution**, denoted by $X \sim \chi^2(\nu)$, occurs when $X \sim \Gamma(k/2, 2)$ and has a PDF of the form

$$f_X(x; \nu) = \frac{1}{2^{\nu/2}\Gamma(\nu/2)} x^{\nu/2-1} e^{-x/2} \quad (13.149)$$

where $\nu > 0$ is the degrees of freedom. The **noncentral χ^2 distribution**, denoted by $X \sim \chi^2(\nu, \lambda)$, has a PDF of the form

$$f_X(x; \nu, \lambda) = \sum_{i=0}^{\infty} \frac{e^{-\lambda/2} (\lambda/2)^i}{i!} f_Y(x; \nu + 2i) \quad (13.150)$$

where $f_Y(x; \nu + 2i)$ is a central χ^2 distribution and $\lambda > 0$ is the non-centrality parameter. The **scaled inverse- χ^2 distribution**, denoted by $X \sim I\chi^2(\tau^2, \nu)$, has a PDF of the form

$$f_X(x; \tau^2, \nu) = \frac{(\tau^2\nu/2)^{\nu/2}}{\Gamma(k/2)x^{k/2-1}} \exp\left(\frac{-\nu\tau^2}{2x}\right) \quad (13.151)$$

where τ^2 is the scale parameter and v is the degrees of freedom. If $X \sim \chi^2(k)$, then $X^{-1} \sim \mathcal{I}\chi^2(1/k, k)$.

The χ^2 distributions are used in detection and analysis of perception systems. An important property of the χ^2 distribution if one has the random variable, Y , given by

$$Y = \sum_{i=1}^k X_i^2 \quad (13.152)$$

where X_1, \dots, X_k are independent, normally distributed random variables with parameters μ_i and $\sigma_i = 1$ for $i = 1, \dots, k$, then $Y \sim \chi^2(k, \lambda)$ with

$$\lambda = \sum_{i=1}^k \mu_i^2 \quad (13.153)$$

Furthermore, if $\mu_1 = \dots = \mu_k = 0$, then $Y \sim \chi^2(k)$.

The **Laplace distribution**, also known as the **double exponential distribution**, denoted by $X \sim \text{Laplace}(\mu, \beta)$, has a PDF of the form

$$f_X(x; \mu, \beta) = \frac{1}{2\beta} \exp\left(-\frac{|x - \mu|}{\beta}\right) \quad (13.154)$$

where μ is the location parameter and $\beta > 0$ is the scale parameter. The **K distribution**, denoted by $X \sim K(\alpha, \lambda)$, has a PDF of the form

$$f_X(x; \alpha, \lambda) = \frac{2\lambda}{\Gamma(\alpha)} (\sqrt{\lambda x})^{\alpha-1} K_{\alpha-1}(2\sqrt{\lambda x}) \quad (13.155)$$

where $K_{\alpha-1}$ is the modified Bessel function of the second kind. The **K distribution** is used in radar, electro-optical, and sonar sensor modeling.

Fundamental Continuous Multivariate Probability Distributions

The **multivariate Gaussian distribution**, denoted as $\vec{X} \sim \mathcal{N}(\vec{\mu}, \Sigma)$, has a joint PDF of the form

$$f_{\vec{X}}(\vec{x}; \vec{\mu}, \Sigma) = \frac{1}{(2\pi)^{n/2} \det(\Sigma)^{0.5}} \exp\left[-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu})\right] \quad (13.156)$$

where $\vec{\mu}$ is the mean and Σ is the covariance and completely characterize the random vector. The PDF can be integrated to obtain the joint Gaussian CDF for which there is no analytical solution. Typically this integration is approximated using numerical methods. A useful property of the multivariate Gaussians is that independence and uncorrelatedness are equivalent, i.e. one implies the other.

Another useful property of the multivariate Gaussian is that Gaussian marginal PDFs are also Gaussian distributed and can be formed by simply dropping the unnecessary elements from $\vec{\mu}$ and Σ . As an example, consider the 2-dimensional case, i.e.,

$$\vec{X} = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix} \quad (13.157)$$

with

$$\vec{\mu} = \begin{bmatrix} \mu_1 \\ \mu_2 \end{bmatrix} \quad (13.158)$$

and

$$\Sigma = \begin{bmatrix} \sigma_1^2 & \sigma_{1,2} \\ \sigma_{1,2} & \sigma_2^2 \end{bmatrix} \quad (13.159)$$

Then, the marginal distribution of $X_1 \sim \mathcal{N}(\mu_1, \sigma_1^2)$ where one has dropped $\mu_2, \sigma_2, \sigma_{1,2}$ and the marginal distribution of $X_2 \sim \mathcal{N}(\mu_2, \sigma_2^2)$ where one has dropped $\mu_1, \sigma_1, \sigma_{1,2}$.

A third useful property of random vectors is for independent multivariate Gaussians $\vec{X} \sim \mathcal{N}(\vec{\mu}_X, \Sigma_X)$ and $\vec{Y} \sim \mathcal{N}(\vec{\mu}_Y, \Sigma_Y)$, their sum is defined as

$$\vec{Z} = \vec{X} + \vec{Y} \quad (13.160)$$

can be shown to be $\vec{Z} \sim \mathcal{N}(\vec{\mu}_X + \vec{\mu}_Y, \Sigma_X + \Sigma_Y)$. However, it should be noted that this is *not* true for dependent Gaussians. Similarly, the affine transformation for multivariate Gaussians, $\vec{Y} = H\vec{X} + \vec{b}$, results in a distribution $\vec{Y} \sim \mathcal{N}(\vec{b} + H\vec{\mu}, H\Sigma H^T)$. Lastly, the **confidence region** for multivariate Gaussian which consists of all vectors, \vec{x} , which satisfy

$$D_M(\vec{x}, X) = (\vec{x} - \vec{\mu})^T \Sigma^{-1} (\vec{x} - \vec{\mu}) \leq F_{\chi_n^2}^{-1}(p) \quad (13.161)$$

where $F_{\chi_n^2}(p)$ is the central chi-squared CDF with n degrees of freedom at the probability p where n is the dimension of \vec{x} . This results from the fact that the sum of squared Gaussians is χ^2 distributed.

Furthermore, by using the eigenvalue decomposition of the covariance, i.e.

$$\Sigma = U\Lambda^{1/2}(U\Lambda^{1/2})^T \quad (13.162)$$

for the multivariate Gaussian $\vec{X} \sim \mathcal{N}(\vec{\mu}, \Sigma)$. Then, by the definition of the PDF, one can show

$$\vec{X} \sim \vec{\mu} + U\mathcal{N}(0, \Lambda) \quad (13.163)$$

or

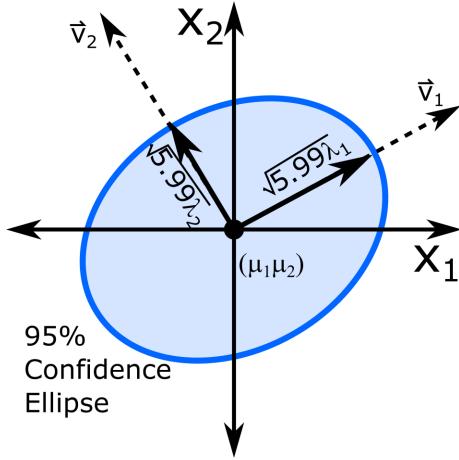
$$\vec{X} \sim \vec{\mu} + U\Lambda^{1/2}\mathcal{N}(0, I) \quad (13.164)$$

where the value of $\Lambda^{1/2}$ define the principle standard deviations and the eigenvector columns of U define the principle axes of the confidence region.

As an example, assume that for a two-dimensional Gaussian random variable, i.e. $\vec{X} = [X_1 \ X_2]^T \sim \mathcal{N}(\vec{\mu}, \Sigma)$ with $\vec{\mu} = [\mu_1 \ \mu_2]^T$, and one can decompose Σ into $V = [\vec{v}_1 \ \vec{v}_2]$ where $\Lambda = \text{diag}(\lambda_1, \lambda_2)$. Then, consider the 95% confidence level, i.e.

$$F_{\chi^2(2)}^{-1}(0.95) = 5.99 = \left(\frac{x_1}{\sqrt{\lambda_1}} \right)^2 + \left(\frac{x_2}{\sqrt{\lambda_2}} \right)^2 \quad (13.165)$$

where the left side defines the **confidence ellipse** along the eigenvector principle axes. The length of each ellipse axis is $2\sqrt{5.99\lambda_1}$ and $2\sqrt{5.99\lambda_2}$. Thus, a 95% error ellipse can be drawn as



The **Gaussian scale mixture (GSM) distribution** is defined for the random vector \vec{Y} as

$$\vec{Y} = \vec{\mu} + \sqrt{Z}\vec{X} \quad (13.166)$$

where $\vec{X}|(Z = z) \sim \mathcal{N}(\mu, z^2\Sigma)$ and $Z > 0$ is the univariate scale parameter and is independent from X .

$$f_Y(x) = \int_0^\infty f_X(x; \mu, z\Sigma) f_Z(z) dz \quad (13.167)$$

With this representation, one can show that the **multivariate symmetric α -stable (MSaS) distribution** occurs for $Z \sim S(\alpha/2, 1, 1, 0)$, the **multivariate t distribution** occurs for $Z \sim \Gamma(\nu/2, 2/\nu)$, and the **multivariate K (MK) distribution** occurs for $Z \sim \Gamma(k, \theta)$.

The **Wishart distribution**, denoted as $\Lambda \sim \mathcal{W}(\Phi, \nu)$, has a joint PDF of the $p(p+1)/2$ elements $\Lambda_{i,j}$ for $i \leq j$ of the form

$$f_\Lambda(\Lambda; \Phi, \nu) = \frac{1}{2^{\nu n/2} |\Phi|^{\nu/2} \Gamma_p(\nu/2)} \det(X)^{(\nu-p-1)/2} \exp\left(-\frac{1}{2} \text{Tr}(\Phi^{-1}\Lambda)\right) \quad (13.168)$$

where $\Lambda \in \mathbb{R}^{p \times p}$ must be symmetric, positive semi-definite, $\Phi \in \mathbb{R}^{p \times p}$ must be symmetric positive definite, and $\nu \geq p - 1$ is the degrees of freedom. If one has N samples of a random vector $\vec{Y} \sim \mathcal{N}(\vec{0}, \Sigma)$, $\vec{y}_1, \dots, \vec{y}_N$ where one can form

$$Y = [\vec{y}_1 \ \cdots \ \vec{y}_N] \quad (13.169)$$

Then, $X = YY^T \sim \mathcal{W}(\Sigma, N)$.

The **inverse-Wishart (IW) distribution**, denoted as $\Sigma \sim \mathcal{IW}(\Phi, \nu)$ has a joint PDF of the form

$$f_\Sigma(\Sigma; \Phi, \nu) = \frac{|\Phi|^{\nu/2}}{2^{\nu p/2} \Gamma_p(\nu/2)} \det(\Sigma)^{-(\nu+p)/2+1} \exp\left(-\frac{1}{2} \text{Tr}(\Phi\Sigma^{-1})\right) \quad (13.170)$$

where $\Sigma \in \mathbb{R}^{p \times p} > 0$ must be symmetric, positive semi-definite, $\Phi \in \mathbb{R}^{p \times p} > 0$ must be symmetric positive definite, $\nu > p - 1$ is the degrees of freedom, $\Gamma_p(a)$ is the **multivariate Gamma function**, i.e.,

$$\Gamma_p(a) = \pi^{p(p-1)/4} \prod_{j=1}^p \Gamma(a + (1-j)/2) \quad (13.171)$$

where if $\Sigma \sim \mathcal{IW}(\Phi, \nu)$, then $\Sigma^{-1} \sim \mathcal{W}(\Phi^{-1}, \nu)$. Notably, the mode of $\mathcal{IW}(\Phi, \nu)$ is $\Phi(\nu + p + 1)^{-1}$ and for $\nu > p + 1$, the mean of $\mathcal{IW}(\Phi, \nu)$ is $\Phi(\nu - p - 1)^{-1}$.

Then, similar to a GSM, the **Gaussian-inverse-Wishart (GIW) distribution**

$$f_{\vec{x}, \Sigma}(\vec{x}, \Sigma; \vec{\mu}, \lambda, \Phi, \nu) = f_N(\vec{x}; \vec{\mu}, \lambda^{-1}\Sigma) f_{\mathcal{IW}}(\Sigma; \Phi, \nu) \quad (13.172)$$

where Σ is the covariance of the conditional Gaussian distribution of $\vec{x} \in \mathbb{R}^n$. For univariate mean μ and variance σ^2 , one can define the **Gaussian-inverse- χ^2 (GIW) distribution**

$$f_{x, \sigma^2}(x, \sigma^2; \mu, \lambda, \sigma^2, \nu) = f_N(x; \mu, \lambda^{-1}\sigma^2) f_{\mathcal{I}\chi^2}(\sigma^2; \tau^2, \nu) \quad (13.173)$$

These distributions are used in perception systems for modeling unknown covariances matrices with known and unknown means in Bayesian estimation.

References

- Nolan, J. P., “1 Basic Properties of Univariate Stable Distributions” in *Stable Distributions – Models for Heavy Tailed Data*, 2017, pp. 3-22
- Eltoft, T., Kim, T., and Lee, T.-W. “Multivariate Scale Mixture of Gaussians Modeling,” in *Independent Component Analysis and Blind Signal Separation*, 2006, pp. 799-806

13.4 Random Processes

When considering a collection of random variables or vectors that are indexed by some mathematical set, one has a **random process**, also known as a **stochastic process** defined as

$$\{\vec{X}(t) : t \in \mathbb{T}\} \quad (13.174)$$

where each random vector $\vec{X}(t)$ is uniquely associated with an element, t , in the **index set**, \mathbb{T} , where \mathbb{T} is often *time*, but can also be a spatial quantity, e.g., images. Furthermore, the values that the random process can take are known as the **state space**, \mathbb{S} , and can be either a continuous or discrete state space. Another important distinction in random processes is the **cardinality classification** where a random process is called continuous-time if \mathbb{T} is an uncountable index set and discrete-time if \mathbb{T} is a countable index set. A discrete-time random process is also known as a **random sequence** and can be distinguished from a continuous-time random process by square brackets, i.e., $\vec{X}[k]$. However, for this section, () will be used for both to explain common concepts.

A **sample** of a random process, \vec{X} , is a *single* outcome at t . For time-indexed processes, a sample is also known as a **time sample**. In contrast, the outcome of the combined samples through the index set \mathbb{T} is the **realization** of the random process, also known as the **sample function**. For time-indexed processes, a realization is also known as the **sample path** or **sample trajectory**. When discussing random processes, the difference between two samples with the random process is called the **increment** whose probability model describes how a random process can change over a certain time period. For any times $t_1 \leq t_2$, the increment is represented by the difference in the indexed random vectors as $\vec{X}(t_2) - \vec{X}(t_1)$.

To describe the probabilities and relationships between elements of a random process from indices $t \in \mathbb{T}$, for a discrete-time random process, one can define the **joint PMF** now indexed by t as

$$p_{\vec{X}}(\vec{x}, t) = \mathbb{P}(\vec{X}(t) = \vec{x}) \quad (13.175)$$

and for a continuous-time random processes, one can define the **joint CDF** now indexed by t as

$$F_{\vec{X}}(\vec{x}, t) = \mathbb{P}(\vec{X}(t) \leq \vec{x}) \quad \forall t \in \mathbb{T} \quad (13.176)$$

as well as the **joint PDF** now indexed by t as

$$f_{\vec{X}}(\vec{x}, t) = \frac{dF_{\vec{X}}(\vec{x}, t)}{d\vec{x}} \quad (13.177)$$

Thus, one can see that the dynamic or spatial nature of the random variable or vector can now be modeled explicitly in the probability functions. The nature of this dynamic/spatial dependence allows for different properties of random processes to be modeled and analyzed beyond random vector theory.

Statistics of Random Processes

To characterize the statistics of a random process with respect to the index $t \in \mathbb{T}$, one can define the **expectation function** which also defines the **mean function**, $\mu_{\vec{X}}(t)$, as

$$\mathbb{E}[\vec{X}(t)] = \mu_{\vec{X}}(t) = \int_{-\infty}^{\infty} \vec{x} f_{\vec{X}}(\vec{x}, t) d\vec{x} \quad (13.178)$$

The **auto-correlation function** between any two samples is defined as

$$R_{\vec{X}\vec{X}}(t_1, t_2) = \mathbb{E}[\vec{X}(t_1)\vec{X}^T(t_2)] \quad (13.179)$$

and it should be noted that for any t , t_1 , and t_2

$$R_{XX}(t, t) = \mathbb{E}[X(t)^2] \geq 0 \quad (13.180)$$

and

$$|R_{XX}(t_1, t_2)| \leq \sqrt{R_{XX}(t_1, t_1)R_{XX}(t_2, t_2)} \quad (13.181)$$

The **auto-covariance function** between any two samples is defined as

$$C_{\vec{X}\vec{X}}(t_1, t_2) = \mathbb{E}\left[\left(\vec{X}(t_1) - \mu_{\vec{X}}(t_1)\right)\left(\vec{X}(t_2) - \mu_{\vec{X}}(t_2)\right)^T\right] \quad (13.182)$$

and it can be shown that

$$C_{\vec{X}\vec{X}}(t_1, t_2) = R_{\vec{X}\vec{X}}(t_1, t_2) - \mu_{\vec{X}}(t_1)\mu_{\vec{X}}^T(t_2) \quad (13.183)$$

and a random process, $\vec{X}(t)$, is **homoscedastic** if $C_{\vec{X}\vec{X}}(t_1, t_2) = C_{\vec{X}\vec{X}}$ is constant and finite for all $t_1, t_2 \in \mathbb{T}$.

For any two random processes, $\vec{X}(t)$ and $\vec{Y}(t)$, the **cross-correlation function** between two samples is defined as

$$R_{\vec{X}\vec{Y}}(t_1, t_2) = \mathbb{E}[\vec{X}(t_1)\vec{Y}^T(t_2)] \quad (13.184)$$

and the **cross-covariance function** between two samples is represented by

$$C_{\vec{X}\vec{Y}}(t_1, t_2) = \mathbb{E} \left[(\vec{X}(t_1) - \mu_{\vec{X}}(t_1)) (\vec{Y}(t_2) - \mu_{\vec{Y}}(t_2)) \right] \quad (13.185)$$

or

$$C_{\vec{X}\vec{Y}}(t_1, t_2) = R_{\vec{X}\vec{Y}}(t_1, t_2) - \mu_{\vec{X}}(t_1) \mu_{\vec{Y}}^T(t_2) \quad (13.186)$$

Properties of Random Processes

A random process, $\vec{X}(t)$, is **independent** if $\forall N \in \mathbb{N}$ and every set of indices, $t_1, \dots, t_N \in T$

$$p_{\vec{X}}(\vec{x}, t_1, \dots, t_N) = p_{\vec{X}}(\vec{x}, t_1) \cdots p_{\vec{X}}(\vec{x}, t_N) \quad \forall \vec{x} \in \mathbb{R}^n \quad (13.187)$$

for a discrete-time process or

$$F_{\vec{X}}(x, t_1, \dots, t_N) = F_{\vec{X}}(x, t_1) \cdots F_{\vec{X}}(x, t_N) \quad \forall \vec{x} \in \mathbb{R}^n \quad (13.188)$$

for a continuous-time process. A random process, $\vec{X}(t)$, has **independent increments** if and only if for every $N \in \mathbb{N}$ and any choice $t_0, t_1, t_2, \dots, t_N \in \mathbb{T}$ with $t_0 < t_1 < t_2 < \dots < t_N$, the random vectors $\vec{X}(t_1) - \vec{X}(t_0)$, $\vec{X}(t_2) - \vec{X}(t_1)$, ..., $\vec{X}(t_N) - \vec{X}(t_{N-1})$ are independent.

A random process, $\vec{X}(t)$, is **strict-sense stationary (SSS)** if each random vector indexed in the process is identically distributed $\forall N \in \mathbb{N} < \infty$. Intuitively, this means that as time passes, the probability distribution for a single sample of a SSS random process remains constant which can often be difficult to prove about a process in reality. A weaker notion of stationarity is **wide-sense stationarity (WSS)**, also known as **covariance stationarity**. The random process, $\vec{X}(t)$, is WSS if $\mu_{\vec{X}}(t) = \mu_{\vec{X}}$ does not depend on t , each $\vec{X}(t)$ has finite correlation, and the covariance of the random variables $\vec{X}(t)$ and $\vec{X}(t + \tau)$ depends only on increment length $\tau \forall t \in \mathbb{T}$. Similar to independence implying uncorrelation, SSS implies WSS, but not vice versa. Thus, for WSS random processes, one can analyze the auto-covariance and auto-correlation functions, e.g., $C_{\vec{X}\vec{X}}(\tau)$, $R_{\vec{X}\vec{X}}(\tau)$, as time-invariant, i.e., only dependent on the difference $\tau = t_2 - t_1$. Notably, for a WSS process, $\vec{X}(t)$, one has

$$R_{\vec{X}\vec{X}}(\tau) = R_{\vec{X}\vec{X}}(-\tau) \quad (13.189)$$

and in the scalar case

$$|R_{XX}(\tau)| \leq R_{XX}(0) \quad (13.190)$$

With these definitions in mind, a WSS random process, $\vec{X}(t)$, is a **white noise process** if its mean function $\mu_{\vec{X}} = \vec{0}$ and its auto-correlation function $R_{\vec{X}\vec{X}}(\tau) = \Sigma \delta(\tau) < \infty$ where $\delta(\tau)$ is the **Dirac delta** function. If $\vec{X}(t)$ is also SSS, then **strong-sense white noise process**. If the discrete-time process $\vec{X}[k]$ is heteroskedastic, i.e., Σ varies with time k , then $\vec{X}[k]$ is **weak white noise**. However, if a WSS random process is not white noise, then it is **colored noise**. Furthermore, the WSS definition allows one to compute the Fourier transform of $R_{\vec{X}\vec{X}}(\tau)$ with respect to τ as the **power spectral density (PSD)** which for continuous-time random processes is

$$S_{\vec{X}\vec{X}}(\omega) = \int_{-\infty}^{\infty} R_{\vec{X}\vec{X}}(\tau) e^{-j\tau\omega} d\tau \quad (13.191)$$

and for discrete-time random processes is

$$S_{\vec{X}\vec{X}}(\omega) = \sum_{\tau=-\infty}^{\infty} R_{\vec{X}\vec{X}}(\tau) e^{-j\omega\tau} \quad (13.192)$$

Recalling the definition of white noise, if $\vec{X}(t)$ is a (weakly) white noise process, then $S_{\vec{X}\vec{X}}(\omega)$ is constant. By the Fourier inversion theorem, for continuous-time random processes, one has

$$R_{\vec{X}\vec{X}}(\tau) = \int_{-\infty}^{\infty} S_{\vec{X}\vec{X}}(f) e^{j\omega\tau} d\omega \quad (13.193)$$

and for discrete-time random processes, one has

$$R_{\vec{X}\vec{X}}(\tau) = \frac{1}{2\pi} \sum_{\omega=-\pi}^{\pi} S_{\vec{X}\vec{X}}(f) e^{j\omega\tau} \quad (13.194)$$

It should be noted that the term “power spectral density” comes from its appearance in the **expected average power**, $P_{XX} = \mathbb{E}[X(t)^2]$, of a scalar WSS random process $X(t)$ and is simply $R_{XX}(0)$ which is the maximum of $R_{XX}(\tau)$. Thus, a discrete-time white noise process has finite power, but a continuous-time white noise process has “infinite” expected average power, an impossible process in reality.

Two random processes, $\vec{X}(t)$ and $\vec{Y}(t)$, with the same index set, $t \in \mathbb{T}$, are **mutually independent** if $\forall N \in \mathbb{N}$ and every set of indices, $t_1, \dots, t_N \in T$, $[\vec{X}(t_1) \dots \vec{X}(t_N)]^T$ and $[\vec{Y}(t_1) \dots \vec{Y}(t_N)]^T$ are independent. Similarly, two random processes, $\vec{X}(t)$ and $\vec{Y}(t)$, with the same index set, $t \in \mathbb{T}$, are **uncorrelated** if

$$C_{\vec{X}\vec{Y}}(t_1, t_2) = \mathbb{E}[(\vec{X}(t_1) - \mu_{\vec{X}}(t_1))(\vec{Y}(t_2) - \mu_{\vec{Y}}(t_2))^T] = 0 \quad \forall t_1, t_2 \in \mathbb{T} \quad (13.195)$$

In addition, two random processes, $\vec{X}(t)$ and $\vec{Y}(t)$, with the same index set, $t \in \mathbb{T}$, are **orthogonal** if

$$R_{\vec{X}\vec{Y}}(t_1, t_2) = \mathbb{E}[(\vec{X}(t_1)\vec{Y}(t_2)^T)] = 0 \quad \forall t_1, t_2 \in \mathbb{T} \quad (13.196)$$

$\vec{X}(t)$ and $\vec{Y}(t)$ are **joint wide-sense stationary (WSS)** if they have the same index set, \mathbb{T} , $\mu_{\vec{X}}(t) = \mu_{\vec{X}}$ and $\mu_{\vec{Y}}(t) = \mu_{\vec{Y}}$ do not depend on t , each $\vec{X}(t)$ and $\vec{Y}(t)$ have finite correlation, and the covariance of the random vectors $\vec{X}(t)$ and $\vec{Y}(t + \tau)$ depends only on increment length $\tau \forall t \in \mathbb{T}$.

Fundamental Random Processes for Perception

The **Markov property** for random processes exists when the current sample of a random process depends only on the immediately previous sample, i.e., the sample at any time is conditionally independent of past values, and future behavior conditionally depends only on the present. Thus, given an initial realization, one can solve for the *probabilities* of achieving any other state at any point in the future. A random process with the Markov property is called a **Markov process**, also known as a **memoryless processes**. An example of a Markov process is a random process with independent increments, but this is not necessary. A Markov process with a countable state space S , is known as a **Markov chain**. A continuous-time Markov chain is also known as a **Markov jump process**. In Markov processes, the sample at any time is called the **state** and the change in the current state to a future state is called a **state transition**.

For discrete-time Markov chains, the countable number of states, $\vec{x}_i \forall i \in \mathbb{N}$ have associated **one-step state transition probabilities** of changing from one state to any other possible future state in the state space \mathbb{S} , i.e.,

$$P_{i,j}[k] = \mathbb{P}(\vec{X}[k+1] = \vec{x}_j | \vec{X}[k] = \vec{x}_i) \quad (13.197)$$

where

$$\sum_j P_{i,j}[k] = 1 \quad (13.198)$$

The (possibly infinite-dimensional) **state transition matrix**, $P[k]$, has the element $P_{i,j}[k]$ at the i^{th} row and the j^{th} column.

If $P[k] = P \forall k$, then the Markov chain is **time-homogeneous** and the m -step **state transition probabilities** defined as

$$P_{i,j}^{(m)} = \mathbb{P}(\vec{X}[k+m] = \vec{x}_j | \vec{X}[k] = \vec{x}_i) \quad (13.199)$$

are stationary and satisfy the **Chapman-Kolmogorov equation (CKE)**

$$P_{i,j}^{(n+m)} = \sum_k P_{i,k}^{(n)} P_{k,j}^{(m)} \quad (13.200)$$

or for the state matrix, $P^{(n+m)}$ containing all $P_{i,j}^{(n+m)}$, one has

$$P^{n+m} = P^n P^m \quad (13.201)$$

Consider the initial condition $v_j = \mathbb{P}(\vec{X}[0] = \vec{x}_j)$.

Then, by the LTP, one has

$$\rho_j^{(n)} = \sum_i \mathbb{P}(\vec{X}[k+m] = \vec{x}_j | \vec{X}[k] = \vec{x}_i) \mathbb{P}(\vec{X}[0] = \vec{x}_i) = \sum_i P_{i,j}^{(n)} v_i \quad (13.202)$$

Then, by defining $\vec{\rho}^{(n)} = [\rho_1^{(n)}, \rho_2^{(n)}, \dots]^T$ and $\vec{v} = [v_1, v_2, \dots]^T$, one has

$$(\vec{\rho}^{(n)})^T = \vec{v}^T P^n \quad (13.203)$$

where if there exists a PMF $\vec{\pi}$ such that

$$(\vec{\rho}^{(n)})^T = \vec{\pi}^T = \vec{\pi}^T P \quad (13.204)$$

Then, $\vec{\pi}$ is a **stationary distribution** and notably defines a left eigenvector of P with normalization $\sum_i \vec{\pi}_i = 1$.

For Markov jump processes, the countable number of states, $\vec{x}_i \forall i \in \mathbb{N}$ have associated **transition probabilities**

$$P_{i,j}(t, s) = \mathbb{P}(\vec{X}(t) = \vec{x}_j | \vec{X}(s) = \vec{x}_i) \quad (13.205)$$

where for time-homogeneous Markov jump process

$$P_{i,j}(t) = \mathbb{P}(\vec{X}(t) = \vec{x}_j | \vec{X}(0) = \vec{x}_i) \quad (13.206)$$

which provides a **Chapman-Kolmogorov equation (CKE)** as

$$P_{i,j}(t+s) = \sum_k P_{i,k}(t) P_{k,j}(s) \quad (13.207)$$

The **transition rate**

$$F_{i,j} = \lim_{\Delta t \downarrow 0} \frac{P_{i,j}(\Delta t)}{\Delta t} \quad (13.208)$$

The **sojourn time**, also known as the **holding time**, is the time spent in one state before jumping to the next state, given as

$$\exp\left(-\lim_{\Delta t \downarrow 0} \frac{P_{i,i}(\Delta t) - 1}{\Delta t}\right) = \exp(-F_{i,i}) \quad (13.209)$$

A Markov jump process $X(t)$ with transition rate $F_{i,i+1} = \lambda$ is a **Poisson process**, also known as a **counting process** if it has the following properties:

- 1 $X(0) = 0$;
- 2 $X(t)$ continuous in $t \geq 0$;
- 3 $X(t)$ has independent increments; and
- 4 $X(t)$ has Poisson increments, i.e.,

$$X(t + \tau) - X(t) \sim \text{Pois}(\lambda\tau) \quad (13.210)$$

Sojourn time for Poisson process is e^λ .

Another important type of continuous- or discrete-time random process is the **Gaussian process**, $\vec{X}(t)$, which for every finite set of indices $t_1, \dots, t_N \in \mathbb{T}$, the finite vector $[\vec{X}(t_1), \dots, \vec{X}(t_N)]^T$ is distributed as a multivariate Gaussian random vector. This also implies that every linear combination of $\vec{X}(t_i)$ is a multivariate Gaussian. An important property of a WSS Gaussian process is that it is also guaranteed to be SSS since the covariance only depends on separation, not individual time values. Furthermore, WSS Gaussian processes are completely defined by their auto-covariance function. $X(t)$ is a **white Gaussian noise (WGN) process** if it is both a Gaussian and a white noise process. $\vec{X}(t)$ is a **Gauss-Markov process** if it is both a Gaussian and a Markov process.

The non-stationary Gauss-Markov process, $X(t)$, is a continuous-time **Wiener process**, also known as the **Brownian motion**, if it has the following properties:

- 1 $X(0) = 0$;
- 2 $X(t)$ continuous in $t \geq 0$;
- 3 $X(t)$ has independent increments; and
- 4 $X(t)$ has Gaussian increments, i.e.,

$$X(t + \tau) - X(t) \sim \mathcal{N}(0, \tau\sigma^2) \quad (13.211)$$

If $\sigma^2 = 1$, then it is a **standard Wiener process** and is the integral of a scalar WGN process.

The only stationary Gauss-Markov process, $X(t)$, is a continuous-time **Ornstein-Uhlenbeck process** with an auto-covariance function as

$$C_{XX}(\tau) = \frac{\sigma^2}{2\theta} \exp(-\theta|\tau|) \quad (13.212)$$

where $\sigma > 0$ is the standard deviation of the noise fluctuations and $\theta > 0$ is the drift which captures how much two points $X(t_1)$ and $X(t_2)$ influence each other. The Ornstein-Uhlenbeck process can be defined via a stochastic differential equation as

$$dX(t) = -\theta X(t)dt + \sigma dW(t) \quad (13.213)$$

where $W(t)$ denotes a standard Wiener process. An Ornstein-Uhlenbeck process is a mean-reverting process as θ is constrained to be positive and connects the idea of stationarity of random processes to stability for stochastic differential equations. If $\theta = 0$, the equation would represent a Wiener process with variance σ^2 .

$X[k]$ is a discrete-time **n^{th} -order auto-regressive (AR) process**, denoted $AR(n)$, if

$$X[k] + a_1 X[k-1] + \cdots + a_n X[k-n] = \epsilon[k] \quad (13.214)$$

where $\epsilon[k]$ is a weakly white noise process, i.e., an uncorrelated sequence of zero-mean random variables with common variance $\sigma^2 = \mathbb{E}[\epsilon[k]]$. Notably, $AR(1)$ is a Markov process and can be considered the discrete-time analogue of the Ornstein-Uhlenbeck process if $|a_1| < 1$.

However, one can extend the Markov property to an arbitrary number of past states using the **n^{th} -order Markov property** for random processes exists when the current sample of a random process depends only on the previous n samples or by the n derivatives of the state. In this context, a first-order Markov process is the traditional Markov process, but multiple time steps allow for $AR(n)$ processes to be n^{th} -order Markov processes of order n . This concept also applies to vectorized continuous-space Markov processes, e.g., **vector auto-regressive (VAR) process**, where the order n is the dimension of the state vector that may be modeled as discrete- or continuous-time. These ideas will be expanded in the next section using stochastic state-space models.

References

For more information, please refer to the following

- Gubner, J.A., “10 Introduction to Random Processes,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 383-442
- Gubner, J.A., “11 Advanced Concepts in Random Processes,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 443-475
- Gubner, J.A., “12 Introduction to Markov Chains,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 476-516
- Gubner, J.A., “15.5 ARMA processes,” in *Probability and Random Processes for Electrical and Computer Engineers*, 1st Ed., Cambridge University Press, 2006, pp. 606-608

- Simon, D., “2.5 Stochastic Processes,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 68-74
- Simon, D., “8.1 Discrete-Time and Continuous-Time White Noise,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 230-232

13.5 Random Dynamical Systems for Control and Perception

A dynamical system is a **random dynamical system** if from a given initial state, the same inputs may produce different outputs. Random dynamical systems are also sometimes referred to as **stochastic dynamical systems** as “stochastic” refers to the modeling approach and “random” refers to the phenomena, but these two terms are often used interchangeably. For random dynamical systems it is typical to use a **stochastic state-space model** which can be defined for continuous-time as

$$\begin{aligned}\dot{\vec{x}}(t) &= f(\vec{x}(t), \vec{u}(t), \vec{w}(t), t) \\ \vec{y}(t) &= h(\vec{x}(t), \vec{v}(t), t)\end{aligned}\tag{13.215}$$

for discrete-time as

$$\begin{aligned}\vec{x}[k+1] &= f(\vec{x}[k], \vec{u}[k], \vec{w}[k], k) \\ \vec{y}[k] &= h(\vec{x}[k], \vec{v}[k], k)\end{aligned}\tag{13.216}$$

and for hybrid-time as

$$\begin{aligned}\dot{\vec{x}}(t) &= f(\vec{x}(t), \vec{u}(t), \vec{w}(t), t) \\ \vec{y}[k] &= h(\vec{x}[k], \vec{v}[k], k)\end{aligned}\tag{13.217}$$

where t is the time, k is the time step, $\Delta t = t/k$ is the time interval, and \vec{w} and \vec{v} are the **process noise** and the **measurement noise**, respectively. In these models, the first stochastic differential or difference function is called the **process equation**, also known as the **state equation**, as it models the state of the random process and the second algebraic function is called the **measurement equation**, also known as the **observation equation**, as it models a random vector transformation of the process state to the observed measurement.

Notably, one can also define a **linear stochastic state-space model** for continuous-time as

$$\begin{aligned}\dot{\vec{x}}(t) &= A(t)\vec{x}(t) + B(t)\vec{u}(t) + L(t)\vec{w}(t) \\ \vec{y}(t) &= C(t)\vec{x}(t) + M(t)\vec{v}(t)\end{aligned}\tag{13.218}$$

for discrete-time as

$$\begin{aligned}\vec{x}[k+1] &= F[k]\vec{x}[k] + G[k]\vec{u}[k] + L[k]\vec{w}[k] \\ \vec{y}[k] &= H[k]\vec{x}[k] + M[k]\vec{v}[k]\end{aligned}\tag{13.219}$$

and for hybrid-time as

$$\begin{aligned}\dot{\vec{x}}(t) &= A(t)\vec{x}(t) + B(t)\vec{u}(t) + L(t)\vec{w}(t) \\ \vec{y}[k] &= H[k]\vec{x}[k] + M[k]\vec{v}[k]\end{aligned}\tag{13.220}$$

where the L is the process noise matrix and M is the measurement noise matrix.

A **continuous-time stochastic hybrid-state-space model** is defined as

$$\begin{aligned}\dot{\vec{x}}(t) &= f(\vec{x}(t), \mathcal{S}(t), \vec{w}(t), t) \\ \vec{y}(t) &= h(\vec{x}(t), \mathcal{S}(t), \vec{v}(t), t)\end{aligned}\quad (13.221)$$

A **discrete-time stochastic hybrid-state-space model** is defined as

$$\begin{aligned}\vec{x}[k+1] &= f_k(\vec{x}[k], \mathcal{S}[k], \vec{w}[k], k) \\ \vec{y}[k] &= h_k(\vec{x}[k], \mathcal{S}[k], \vec{v}[k], k)\end{aligned}\quad (13.222)$$

A **hybrid-time stochastic hybrid-state-space model** is defined as

$$\begin{aligned}\dot{\vec{x}}(t) &= f(\vec{x}(t), \mathcal{S}(t), \vec{w}(t), t) \\ \vec{y}[k] &= h_k(\vec{x}[k], \mathcal{S}[k], \vec{v}[k], k)\end{aligned}\quad (13.223)$$

where \vec{x} is the **base state** which varies continuously, $\mathcal{S} \in \mathbb{S}$ is the **modal state** which is a jump process, i.e., it may either jump discretely or stay constant, \vec{y} is the measurement, \vec{w} is the process noise, and \vec{v} is the measurement noise. Here, the whole state, $\vec{\xi} = (\vec{x}, \mathcal{S})$, is a **hybrid process**, and is a **Markov jump system** if \mathcal{S} is a Markov chain/jump process. Often, \mathcal{S} is assumed to be a time-homogeneous Markov process.

A **jump-linear system** for continuous-time is

$$\begin{aligned}\dot{\vec{x}}(t) &= A(\mathcal{S}(t))\vec{x}(t) + B(\mathcal{S}(t))\vec{u}(t) + L(\mathcal{S}(t))\vec{w}(t) \\ \vec{y}(t) &= C(\mathcal{S}(t))\vec{x}(t) + M(\mathcal{S}(t))\vec{v}(t)\end{aligned}\quad (13.224)$$

and for discrete-time is

$$\begin{aligned}\vec{x}[k+1] &= F(\mathcal{S}[k])\vec{x}[k] + G(\mathcal{S}[k])\vec{u}[k] + L(\mathcal{S}[k])\vec{w}[k] \\ \vec{y}[k] &= H(\mathcal{S}[k])\vec{x}[k] + M(\mathcal{S}[k])\vec{v}[k]\end{aligned}\quad (13.225)$$

where if \mathcal{S} is a Markov jump process, this is a **Markov jump-linear system (MJLS)**. This system is generally nonlinear, but may be considered a type of linear, parameter-varying (LPV) system where \mathcal{S} is the parameter.

Thus, stochastic state-space models are functions of the random processes/sequences and require an initial prior *distribution* for $\vec{x}(0)$ or $\vec{x}[0]$ to solve. It is typical to assume in discrete-time $\vec{w}[k]$ and $\vec{v}[k]$ are white noise processes, i.e., they are zero-mean with no auto-correlation for different k . In this case, the process equation of the discrete-time linear stochastic state-space model is also known as a **vector auto-regressive (VAR) process**. Also, it is typical to assume that for continuous-time, $\vec{w}(t)$ and $\vec{v}(t)$ having independent “increments” for infinitesimal $d\vec{w}(t)$ or $d\vec{v}(t)$, i.e., are n^{th} -order Markov processes. Under these assumptions of the n^{th} -order Markov property for \vec{X} where one does not observe the state \vec{X} directly, but only “knows” \vec{u} and \vec{y} , these stochastic state-space systems are types of **Hidden Markov Models (HMM)**. Lastly, one often assumes that $x(0)$, \vec{w} , and \vec{v} are independent of each other as well, or at least uncorrelated. For the simplest model of the stochastic noise, assume that $\vec{w}[k]$, $\vec{v}[k]$, $d\vec{w}(t)$, and $d\vec{v}(t)$ are vectorized additive white Gaussian noises (AWGN), i.e., $\vec{w}(t)$ and $\vec{v}(t)$ are vectorized additive Wiener processes as integrals of the white Gaussian noise process. This type of model is often referred to as “natural” noise because it arises in several cases, e.g., thermal vibrations of atoms in conductors, black-body radiation, and deep space signals.

Stochastic Optimal Control

Using observer-based feedback control for stochastic state-space systems, one can form the stochastic control systems, including stochastic eigenvalue placement and stochastic optimal control. In this context, the **fundamental stochastic optimal control problem** known as the **linear-quadratic-Gaussian (LQG) OCP** linear-quadratic Gaussian OCP defined for continuous-, discrete-, and hybrid-time. The LQG term derives from the *linear* process equation, measurement equation, and additive noise, *quadratic* due to the form of the cost function/functional, and having white *Gaussian* noise with a constant covariance matrix. For the cases below, assume the system is controllable and observable. Recall that in all three cases, the design of the state feedback and state estimator gains independently due to the **separation principle**.

The unconstrained finite-horizon continuous-time LQG OCP is

$$\begin{aligned} \vec{u}^{opt}(t) = & \underset{\vec{u}(t) \text{ for } t \geq 0}{\operatorname{argmin}} \quad \mathcal{J} = \mathbb{E} \left[\vec{x}^T(t_f) E \vec{x}(t_f) + \int_0^{t_f} \vec{x}^T(t) Q(t) \vec{x}(t) + \vec{u}^T(t) R(t) \vec{u}(t) \right] \\ \text{subject to: } & \dot{\vec{x}}(t) = A(t) \vec{x}(t) + B(t) \vec{u}(t) + L(t) \vec{w}(t) \\ & \vec{y}(t) = C(t) \vec{x}(t) + M(t) \vec{v}(t) \\ \text{initial condition: } & \vec{\mu}_0 = \mathbb{E}[\vec{x}(0)], \Sigma_0 = \mathbb{E}[\vec{x}(0) \vec{x}^T(0)] \end{aligned} \quad (13.226)$$

where $d\vec{w}(t)$ and $d\vec{v}(t)$ are AWGN with covariances, $\Sigma_{\vec{w}}$, and $\Sigma_{\vec{v}}$, respectively.

To solve this OCP, assume one can use an observer-based feedback controller, i.e.

$$\vec{u}(t) = -K_{LQR}(t) \hat{\vec{x}}(t) \quad (13.227)$$

where $K_{LQR}(t)$ is the optimal continuous-time LQR gain matrix. The initial condition of the estimator is given by an initial estimate of the state, $\hat{\vec{x}}(t) = \vec{\mu}_0$ and initial covariance, Σ_0 . The state estimate $\hat{\vec{x}}(t)$ evolves according to the **continuous-time Luenberger observer equation** as

$$\dot{\hat{\vec{x}}}(t) = A(t) \hat{\vec{x}}(t) + B(t) \vec{u}(t) + K_{LQE} \left(\vec{y}(t) - C(t) \hat{\vec{x}}(t) \right) \quad (13.228)$$

For this stochastic OCP, the optimal estimator gain, $K_{LQE}(t)$, is the continuous-time **linear-quadratic estimator (LQE)**, also known as the **continuous-time Kalman gain**.

Recall that to find the optimal $K_{LQR}(t)$, one must solve the corresponding Riccati differential equation backwards-in-time for the closed-loop dynamics of the state controller, i.e.

$$\dot{P}_{LQR}(t) = -P_{LQR}(t)A(t) - A^T(t)P_{LQR}(t) + P_{LQR}(t)B(t)R^{-1}(t)B^T(t)P_{LQR}(t) - Q(t) \quad (13.229)$$

with final condition

$$P_{LQR}(t_f) = E \quad (13.230)$$

Then, solving for $P_{LQR}(t)$ provides the optimal LQR gain as

$$K_{LQR}(t) = R^{-1}(t)B^T(t)P_{LQR}(t) \quad (13.231)$$

Similarly, to find the optimal $K_{LQE}(t)$, one must solve the corresponding Riccati differential equation forwards-in-time for the closed-loop dynamics of the state estimator, i.e.

$$\begin{aligned} \dot{P}_{LQE}(t) = & P_{LQE}(t)A(t) + P_{LQE}(t)A^T(t) + L(t)\Sigma_{\vec{w}}L^T(t) \\ & - P_{LQE}(t)C^T(t)(M(t)\Sigma_{\vec{v}}M^T(t))^{-1}(t)C(t)P_{LQE}(t) \end{aligned} \quad (13.232)$$

with initial condition

$$P_{LQE}(0) = \Sigma_0 \quad (13.233)$$

Then, solving for P_{LQE} provides the LQE gain as

$$K_{LQE}(t) = P_{LQE} C^T(t) (M(t) \Sigma_{\vec{v}} M^T(t))^{-1} \quad (13.234)$$

Note that the unconstrained infinite-horizon continuous-time LQG OCP would require solving two CAREs and provide constant steady-state K_{LQR} and K_{LQE} .

The unconstrained finite horizon discrete-time LQG OCP is

$$\begin{aligned} \vec{u}^{opt}[k] = \underset{\vec{u}[k] \text{ for } k=0, \dots, N-1}{\operatorname{argmin}} \quad & \mathcal{J} = \mathbb{E} \left[\vec{x}^T[N] E \vec{x}[N] + \sum_{k=0}^{N-1} \vec{x}^T[k] Q[k] \vec{x}[k] + \vec{u}^T[k] R[k] \vec{u}[k] \right] \\ \text{subject to: } & \vec{x}[k+1] = F[k] \vec{x}[k] + G[k] \vec{u}[k] + L[k] \vec{w}[k] \\ & \vec{y}[k] = H[k] \vec{x}[k] + M[k] \vec{v}[k] \\ \text{initial condition: } & \vec{\mu}_0 = \mathbb{E}[\vec{x}[k]], \Sigma_0 = \mathbb{E}[\vec{x}[k] \vec{x}^T[k]] \end{aligned} \quad (13.235)$$

where $\vec{w}[k]$ and $\vec{v}[k]$ are AWGN with covariances, $\Sigma_{\vec{w}}$ and $\Sigma_{\vec{v}}$, respectively.

To solve this OCP, assume one can use an observer-based feedback controller, i.e.

$$\vec{u}[k] = -K_{LQR}[k] \hat{\vec{x}}[k] \quad (13.236)$$

where $K_{LQR}[k]$ is the optimal discrete-time LQR gain matrix. The initial condition of the estimator is given by an initial estimate of the state, $\hat{\vec{x}}[0] = \vec{\mu}_0$ and initial covariance, Σ_0 . Recall that the state estimate $\hat{\vec{x}}[k]$ evolves according to the **discrete-time Luenberger observer equation**

$$\hat{\vec{x}}[k+1] = F[k] \hat{\vec{x}}[k] + G[k] \vec{u}[k] + L \left(\vec{y}[k] - H[k] \hat{\vec{x}}[k] \right) \quad (13.237)$$

where for this stochastic OCP, the optimal estimator gain, $K_{LQE}[k]$, is the discrete-time **linear-quadratic estimator (LQE)**, also known as the **discrete-time Kalman gain**.

Recall that to find the optimal $K_{LQR}[k]$, one must solve the corresponding Riccati difference equation backwards-in-time for the closed-loop dynamics of the state controller, i.e.

$$\begin{aligned} P_{LQR}[k] = & F^T[k] P_{LQR}[k+1] F[k] + Q[k] \\ & - F^T[k] P_{LQR}[k+1] G[k] \left(G^T[k] P_{LQR}[k+1] G[k] + R[k] \right)^{-1} \\ & G^T[k] P_{LQR}[k+1] F[k] \end{aligned} \quad (13.238)$$

with final condition

$$P_{LQR}[k] = E \quad (13.239)$$

Then, solving for $P_{LQR}[k]$ provides the optimal LQR gain as

$$K_{LQR}[k] = \left(G^T[k] P_{LQR}[k+1] G[k] + R[k] \right)^{-1} G^T[k] P_{LQR}[k+1] F[k] \quad (13.240)$$

Similarly, to find the optimal $K_{LQE}[k]$, one must solve the corresponding Riccati difference equation forwards-in-time for the closed-loop dynamics of the state estimator, i.e.

$$\begin{aligned} P_{LQE}[k+1] = & F[k]P_{LQE}[k]F^T[k] + L[k]\Sigma_{\vec{w}}L^T[k] \\ & - F[k]P_{LQE}[k]H^T[k] \left(H[k]P_{LQE}[k]H^T[k] + M[k]\Sigma_{\vec{v}}M^T[k] \right)^{-1} \\ & H[k]P_{LQE}[k]F^T[k] \end{aligned} \quad (13.241)$$

with initial condition

$$P_{LQE}[0] = \Sigma_0 \quad (13.242)$$

Then, solving for P_{LQE} provides the LQE gain as

$$K_{LQE}[k] = P_{LQE}[k]H^T[k] \left(H[k]P_{LQE}[k]H^T[k] + M[k]\Sigma_{\vec{v}}M^T[k] \right)^{-1} \quad (13.243)$$

Note, that the unconstrained infinite-horizon discrete-time LQG OCP would require solving two DAREs and provide constant steady-state K_{LQR} and K_{LQE} .

The unconstrained finite-horizon hybrid-time LQG OCP is

$$\begin{aligned} \vec{u}^{opt}(t) = & \underset{\vec{u}(t) \text{ for } t \geq 0}{\operatorname{argmin}} \mathcal{J} = \mathbb{E} \left[\vec{x}^T(t_f)E\vec{x}(t_f) + \int_0^{t_f} \vec{x}^T(t)Q(t)\vec{x}(t) + \vec{u}^T(t)R(t)\vec{u}(t) \right] \\ \text{subject to: } & \dot{\vec{x}}(t) = A(t)\vec{x}(t) + B(t)\vec{u}(t) + L(t)\vec{w}(t) \\ & \vec{y}[k] = H[k]\vec{x}[k] + M[k]\vec{v}[k] \\ \text{initial condition: } & \vec{\mu}_0 = \mathbb{E}[\vec{x}(0)], \Sigma_0 = \mathbb{E}[\vec{x}(0)\vec{x}^T(0)] \end{aligned} \quad (13.244)$$

where $d\vec{w}(t)$ and $\vec{v}[k]$ are AWGN with covariances, $\Sigma_{\vec{w}}$ and $\Sigma_{\vec{v}}$, respectively, $t = k\Delta t \forall k \in \mathbb{N}$.

To solve this OCP, assume one can use an observer-based feedback controller, i.e.

$$\vec{u}(t) = -K_{LQR}(t)\hat{\vec{x}}(t) \quad (13.245)$$

where $K_{LQR}(t)$ is the optimal continuous-time LQR gain matrix. The initial condition of the estimator is given by an initial estimate of the state, $\hat{\vec{x}}(t) = \vec{\mu}_0$, and initial covariance, $\Sigma[0]$. The state estimate $\hat{\vec{x}}(t)$ evolves according to the **hybrid-time Luenberger observer equation**.

$$\hat{\vec{x}}(t) = \begin{cases} A(t)\hat{\vec{x}}(t) + B(t)\vec{u}(t) + K_{LQE}[k] \left(\vec{y}[k] - H[k]\hat{\vec{x}}(t) \right) & \text{if } t = k\Delta t \forall k \in \mathbb{N} \\ A(t)\hat{\vec{x}}(t) + B(t)\vec{u}(t) & \text{otherwise} \end{cases} \quad (13.246)$$

For this stochastic OCP, the optimal estimator gain, $K_{LQE}[k]$, is the hybrid-time **linear-quadratic estimator (LQE)**, also known as the **hybrid-time Kalman gain**.

Recall that to find the optimal $K_{LQR}(t)$, one must solve the corresponding Riccati differential equation backwards-in-time for the closed-loop dynamics of the state controller, i.e.

$$\dot{P}_{LQR}(t) = -P_{LQR}(t)A(t) - A^T(t)P_{LQR}(t) + P_{LQR}(t)B(t)R^{-1}(t)B^T(t)P_{LQR}(t) - Q(t) \quad (13.247)$$

with final condition

$$P_{LQR}(t) = E \quad (13.248)$$

Then, solving for $P_{LQR}(t)$ provides the optimal LQR gain as

$$K_{LQR}(t) = R^{-1}(t)B^T(t)P_{LQR}(t) \quad (13.249)$$

Similarly, to find the optimal $K_{LQE}[k]$, one must solve the corresponding k Riccati differential equations forwards-in-time for the closed-loop dynamics of the state estimator, i.e.

$$\dot{P}_{LQE}(\Delta t) = \int_0^{\Delta t} P_{LQE}(\tau)A(\tau) + P_{LQE}(\tau)A^T(\tau) + L(\tau)\Sigma_{\vec{w}}L^T(\tau)d\tau \quad \forall t \in [(k-1)\Delta t, k\Delta t] \quad (13.250)$$

with initial conditions as

$$P_{LQE}(0) = \begin{cases} \Sigma_0 & k = 0 \\ P_{LQE}[k-1] - P_{LQE}[k-1]H^T[k-1] \\ \left(H[k-1]P_{LQE}[k-1]H^T[k-1] + M[k-1]\Sigma_{\vec{v}}M^T[k-1] \right)^{-1} H[k-1]P_{LQE}[k-1] & k \geq 1 \in \mathbb{N} \end{cases} \quad (13.251)$$

Then, solving for $P_{LQE}[k]$ as the k^{th} solution to $\dot{P}_{LQE}(\Delta t)$ provides the LQE gain as

$$K_{LQE}[k] = P_{LQE}[k]H^T[k] \left(H[k]P_{LQE}[k]H^T[k] + M[k]\Sigma_{\vec{v}}M^T[k] \right)^{-1} \quad (13.252)$$

Note that if the state-space model is LTI, then one can compute the discretized matrices, F and L , and use those as a Riccati difference equation for solving for $P_{LQE}[k]$ as

$$\begin{aligned} P_{LQE}[k+1] = & F P_{LQE}[k] F^T + L \Sigma_{\vec{w}} L^T \\ & - F P_{LQE}[k] H^T[k] \left(H[k]P_{LQE}[k]H^T[k] + M[k]\Sigma_{\vec{v}}M^T[k] \right)^{-1} H[k]P_{LQE}[k]F^T \end{aligned} \quad (13.253)$$

Then, the unconstrained infinite-horizon hybrid-time LQG OCP would require solving the CARE and the DARE to provide constant steady-state K_{LQR} and K_{LQE} .

To summarize these results, the LQG OCP solution is **separable** as it uses the optimal LQR gain and the Kalman gain for stochastic dynamical systems. The solution is obtained by solving the **dual** OCPs, i.e. solving the K_{LQE} Riccati equation forward-in-time from the initial condition and the K_{LQR} Riccati equation backward-in-time from the final condition. Also, note that at each step, the Kalman filter computes the state estimate \hat{x} using past \vec{y} and \vec{u} and the feedback controller generates \vec{u} using \hat{x} separately. Although no longer optimal, this framework of state feedback control separate from state estimation is considered as the primary design approach in the aerospace vehicle control and perception systems presented in this textbook even when the stochastic state-space model is nonlinear and non-Gaussian. Lastly, it should be mentioned that LQG OCPs can also be developed for MJLS, but these results are beyond the scope of this textbook.

Sensor Modeling for Perception Systems

Perception systems are used to achieve the **perception** of the dynamical system state and/or its environment via sensor systems and information fusion where the sensor systems may be single or multi-sensor. A **sensor** is devices that detects or measures an object, area, phenomenon, or process in the system state or the

external environment and produces an output signal. Perception systems process the single or multi-sensor output signals into relevant information for planning and control which may include multi-sensor information fusion. As modern sensor systems are primarily implemented with digital computer systems, this textbook will exclusively use discrete-time models for perception systems.

One classification of sensor type is for sensors that detect or measure an object, area, phenomenon, or process in the system state directly are known as **proprioceptive** as opposed to sensors that detect or measure an object, area, phenomenon, or process in the external environment state directly and the system state indirectly are known as **exteroceptive**. A subset of exteroceptive sensing is the additional field of **remote sensing** defined as the science and techniques of obtaining information about an object, area, phenomenon, or process acquired by a sensor that is *not in contact* with the object, area, phenomenon, or process under investigation. Another classification of sensor type is for a **corrective sensor** that directly or indirectly measure the system state itself as opposed to a **predictive sensor** that directly or indirectly measures the changes in the system state.

For corrective sensor data at time step k , $\vec{y}[k]$, one can typically model its probability distribution based on a sample path of n previous time steps and the current sample, i.e., $\vec{x}[k-n], \dots, \vec{x}[k]$, through a conditional PDF

$$f_{\vec{Y}|\vec{X}}(\vec{y}[k]|\vec{x}[k], \dots, \vec{x}[k-n]) \quad (13.254)$$

However, if the measurement is only independent of past states, i.e.,

$$f_{\vec{Y}|\vec{X}}(\vec{y}[k]|\vec{x}[k]) \quad (13.255)$$

then one can alternatively represent this sensor as the measurement equation in a stochastic state-space model, i.e.

$$\vec{y}[k] = h_k(\vec{x}[k], \vec{v}[k]) \quad (13.256)$$

where $\vec{v}[k]$ is the stochastic **corrective sensor error** at time step k .

For predictive sensor data at the previous time step $k-1$, $\vec{u}[k-1]$, one can relate its “state-transition” information with the current state, $\vec{x}[k]$, and the *previous n* states, $\vec{x}[k-1], \dots, \vec{x}[k-n]$, through a conditional PDF

$$f_{\vec{X}}(\vec{x}[k]|\vec{x}[k-1], \vec{u}[k-1], \dots, \vec{x}[k-n], \vec{u}[k-n]) \quad (13.257)$$

However, if the state has a n_x^{th} -order Markov property, i.e.,

$$f_{\vec{X}}(\vec{x}[k]|\vec{x}[k-1], \vec{u}[k-1]) \quad (13.258)$$

then one can alternatively represent as the process equation in a stochastic state-space model, i.e.,

$$\vec{x}[k] = f_{k-1}(\vec{x}[k-1], \vec{u}[k-1], \vec{w}[k-1]) \quad (13.259)$$

where $\vec{w}[k-1]$ is the stochastic **predictive sensor error** at time step $k-1$.

Thus, the use of stochastic state-space systems for multi-sensor information fusion utilizes the same theoretic notions of state estimation seen in the fundamental optimal control problem (OCP). Further study of Kalman filtering for multi-sensor information fusion in perception systems is developed in chapter 15 of this textbook. The consideration of the Kalman filter there will highlight important aspects of the filter beyond the Riccati equations development here.

References

For more information, please refer to the following

- Gelb, A., Kasper, J. F., Nash, R. A., Price, C. F., and Sutherland, A. A, “4.6 Solution of the Riccati Equation,” in *Applied Optimal Estimation*, The M.I.T. Press, 1974, pp. 136-141
- Gelb, A., Kasper, J. F., Nash, R. A., Price, C. F., and Sutherland, A. A, “4.7 Statistical Steady State - The Wiener Filter,” in *Applied Optimal Estimation*, The M.I.T. Press, 1974, pp. 142-155
- Gelb, A., Kasper, J. F., Nash, R. A., Price, C. F., and Sutherland, A. A, “9.5 Optimal Control of Linear Systems,” in *Applied Optimal Estimation*, The M.I.T. Press, 1974, pp. 356-365
- Sarkka, S., “4.1 Probabilistic state space models,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 51-53
- Simon, D., “5.3 One-Step Kalman Filter Equations,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 131
- Simon, D., “8.5.3 Duality,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 258-259

13.6 Detection and Hypothesis Testing Theory

Decision theory is the determination of the optimal decision given constraints and assumptions on a decision problem. This textbook has already looked at optimal control and optimal estimation as special cases of decision theory. **Statistical inference** is a special case of the decision problem and can be stated as deciding which probability distribution one should select to best represent data among some **candidate set** of statistical models, \mathcal{H} . Using the observed data and models, one can generally forms some mathematical law to select the optimal statistical model which can be either fully parametric, non-parametric, or semi-parametric. Fully parametric statistical inference assumes the data samples are *fully* described by a family of PDFs involving only a finite number of unknown parameters, $\vec{\theta}$. Thus, the fully parametric statistical inference problem can be formulated as a type of parameter estimation problem for the PDF parameters, $\hat{\vec{\theta}}$, which may or may not be constrained.

An alternative to the parameter estimation approach is if \mathcal{H} contains a finite number of candidate models known as **hypotheses**, then one can alternatively consider this a statistical comparison problem between hypotheses, i.e. a **hypothesis testing problem** to select a single optimal hypothesis. When \mathcal{H} contains more than two hypothesis, one has a multiple hypothesis testing problem, also known as a **multiple testing problem** or a **multiple comparisons problem**. In either case, the finite hypotheses can be distinguished as \mathcal{H}_i for $i = 0, 1, \dots, M$, where \mathcal{H}_0 is known as the **null hypothesis**, and \mathcal{H}_i for $i = 1, \dots, M$ are the **alternative hypotheses**. From the data, one computes a single value called the **test statistic** and selects an alternative hypothesis if the test statistic exceeds some threshold probability, i.e. some **significance level**, α . Otherwise, one selects the null hypothesis with some **confidence level**, $\gamma = 1 - \alpha$. Some common test statistics for fully parametric hypotheses include Z-tests, t-tests, χ^2 -tests, F-tests, likelihood-ratio tests, and Bayesian tests.

The selection of the null or the alternative hypotheses may be either the correct or erroneous selection. In particular, a **Type I error** occurs when one selects any \mathcal{H}_i for $i = 1, \dots, M$ when the \mathcal{H}_0 is actually true

and a **Type II error** occurs when one selects \mathcal{H}_0 when one of the \mathcal{H}_i for $i = 1, \dots, M$ is actually true. An alternative perspective to hypothesis testing, is known as **detection theory** where one simply declares if an information-bearing pattern, i.e. a *signal*, has occurred within random patterns that distract from the information, i.e. *noise*. In this case, the null hypothesis is defined as *no signal occurring* and the M alternative hypotheses are the occurrence of the m signals. Thus, in detection theory, the test statistic is also known as the **detector**, a Type I error is also known as a **false alarm**, i.e. one declared a signal occurred when no signal actually occurred, and a Type II error is also known as a **missed detection**, i.e. one declared no signal occurred when a signal did actually occur. Thus, in detection, one typically chooses the detector function, D_α , and **detection threshold**, τ , to balance the probability of false alarm, $P_{FA} = \alpha$, with the probability of missed detection, $P_{MD} = \beta$.

Common Null Hypothesis Tests

The **Z-test** is an hypothesis test for which the distribution of the test statistic for the sample set y_1, \dots, y_N , under the null hypothesis is modeled as a normal distribution with expected value, μ_0 , and variance, σ_0^2 , which is known for the Z-test. Often, the test statistic for this test uses the sum of the N IID random variables, e.g. it tests for the mean of the distribution, which by the central limit theorem for finite variance distributions will approach the normal distribution as $N \rightarrow \infty$. In this case, one can set up three different hypothesis tests

1. **Lower-tailed** where \mathcal{H}_0 is $\mu \geq \mu_0$ and \mathcal{H}_1 is $\mu \leq \mu_0$
2. **Upper-tailed** where \mathcal{H}_0 is $\mu \leq \mu_0$ and \mathcal{H}_1 is $\mu \geq \mu_0$
3. **Two-tailed** where \mathcal{H}_0 is $\mu = \mu_0$ and \mathcal{H}_1 is $\mu \neq \mu_0$

For the Z-test, the test statistic, also known as the **standard score** is calculated as

$$T = Z = \left(\frac{\hat{\mu} - \mu_0}{\sigma_0} \right) \quad (13.260)$$

Then, for the lower-tailed hypothesis test

$$F_N(T; 0, 1) \stackrel{\mathcal{H}_1}{\underset{\mathcal{H}_0}{\gtrless}} \alpha \quad (13.261)$$

for the upper-tailed hypothesis test

$$F_N(-T; 0, 1) \stackrel{\mathcal{H}_1}{\underset{\mathcal{H}_0}{\gtrless}} \alpha \quad (13.262)$$

and for the two-tailed hypothesis test

$$2F_N(-|T|; 0, 1) \stackrel{\mathcal{H}_1}{\underset{\mathcal{H}_0}{\gtrless}} \alpha \quad (13.263)$$

where $F_N(x; 0, 1)$ is the CDF of the standard normal distribution, i.e. a zero-mean Gaussian with unit variance, α is the significance of the test, and the double inequality denotes the selection of \mathcal{H}_0 or \mathcal{H}_1 depending on the values on the left side relative to α .

The **t-test** is an hypothesis test for which the distribution of the test statistic for the sample set y_1, \dots, y_N , under the null hypothesis is modeled as a *t*-distribution with degrees-of-freedom (DOF), v . Often, the test statistic for this test uses the sum of the N IID random variables, e.g. it tests for the mean of the distribution,

which by the central limit theorem for finite variance distributions will approach the normal distribution as $N \rightarrow \infty$. However, if the sample set has an unknown variance and one must use the variance of the sample set as an estimate of the variance which leads to the t -distribution. In this case, one can set up three different hypothesis tests

1. **Lower-tailed** where \mathcal{H}_0 is $\mu \geq \mu_0$ and \mathcal{H}_1 is $\mu \leq \mu_0$
2. **Upper-tailed** where \mathcal{H}_0 is $\mu \leq \mu_0$ and \mathcal{H}_1 is $\mu \geq \mu_0$
3. **Two-tailed** where \mathcal{H}_0 is $\mu = \mu_0$ and \mathcal{H}_1 is $\mu \neq \mu_0$

For the t -test, the test statistic is calculated as

$$T = \frac{\hat{\mu} - \mu_0}{\frac{\hat{\sigma}}{\sqrt{n}}} \quad (13.264)$$

which is similar to the standard score, but with an estimate of the sample variance, $\frac{\hat{\sigma}^2}{n}$, instead of the true variance.

Then, for the lower-tailed hypothesis test

$$F_t(t; n-1) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \alpha \quad (13.265)$$

for the upper-tailed hypothesis test

$$F_t(-t; n-1) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \alpha \quad (13.266)$$

and for the two-tailed hypothesis test

$$2F_t(-|t|; n-1) \underset{\mathcal{H}_0}{\overset{\mathcal{H}_1}{\gtrless}} \alpha \quad (13.267)$$

where $F_t(x; \nu)$ is the CDF of the t -distribution with ν DOF and α is the significance of the test.

The χ^2 -**test** is an hypothesis test for which the distribution of the test statistic under the null hypothesis is modeled as a (central) χ^2 -distribution with degrees-of-freedom (DOF), k . It can be shown that the χ^2 -distribution occurs for the sum of k squared independent standard normal random variables, i.e. Y is χ^2 distributed with k DOF if

$$Y = \sum_{i=1}^k Z_i \quad (13.268)$$

where Z_1, \dots, Z_k standard normal random variables. Often, the test statistic for this test assumes the sample distribution is normally distributed, but one desires to test for a specific variance for the underlying distribution. In this case, one can set up a hypothesis test with \mathcal{H}_0 is $\sigma^2 \leq \sigma_0^2$ and \mathcal{H}_1 is $\sigma^2 > \sigma_0^2$. For this case, the test statistic for the χ^2 -test is calculated as

$$T = \sum_{i=1}^N \left(\frac{y_i - \bar{y}}{\sigma_0} \right)^2 \quad (13.269)$$

where the mean of the sample is

$$\bar{y} = \frac{1}{N} \sum_{i=1}^N y_i \quad (13.270)$$

Then, for upper-tailed hypothesis test

$$F_{\chi^2}(T; n - 1) \stackrel{\mathcal{H}_1}{\underset{\mathcal{H}_0}{\gtrless}} 1 - \alpha \quad (13.271)$$

where $F_{\chi^2}(x; k)$ is the CDF of the χ^2 -distribution with k DOF and α is the significance of the test.

The **F-test** is an hypothesis test for which the distribution of the test statistic under the null hypothesis is modeled as an F-distribution with two degrees-of-freedom (DOF), d_1 and d_2 . The F-distribution occurs for the ratio of χ^2 random variables with DOFs, d_1 , and d_2 , i.e. X is F distributed with d_1 and d_2 DOFs if

$$X = \frac{\frac{Y_1}{d_1}}{\frac{Y_2}{d_2}} \quad (13.272)$$

where Y_1 and Y_2 are χ^2 random variables with DOFs, d_1 and d_2 , respectively. Note that these χ^2 -distributed random variables are often two separate sets of the sum of squared independent standard normal variables. Often, the test statistic for this tests assume the sample distributions are Gaussian distributed with zero-mean and two variances, σ_1^2 and σ_2^2 , but one desires to test if the variances of the two sets of samples are the same. In this case, one can set up a hypothesis test with \mathcal{H}_0 is $\sigma_1^2 = \sigma_2^2$ and \mathcal{H}_1 is $\sigma_1^2 \neq \sigma_2^2$. For this case, the test statistic for the F-test is calculated as

$$T = \frac{\frac{Y_1}{d_1}}{\frac{Y_2}{d_2}} \quad (13.273)$$

Then, for upper-tailed hypothesis test

$$F_F(T; \sigma_1^2, \sigma_2^2) \stackrel{\mathcal{H}_1}{\underset{\mathcal{H}_0}{\gtrless}} 1 - \alpha \quad (13.274)$$

where $F_F(x; d_1, d_2)$ is the CDF of the F-distribution with d_1 and d_2 DOFs and α is the significance of the test.

Likelihood-Ratio Tests

A fundamental result in detection theory states that for data \vec{y} sampled from random vector \vec{Y} with PDF or likelihood dependent on scalar parameter θ , i.e. $f_{\vec{Y}|\Theta}(\vec{y}|\theta) = \mathcal{L}(\theta)$. Consider a null hypothesis \mathcal{H}_0 as $\vec{\theta} \in \mathcal{T}_0$ and the alternative hypothesis \mathcal{H}_1 as $\theta \in \mathcal{T}_1$, where \mathcal{T}_1 is in the complement of \mathcal{T}_0 . Then, the **likelihood-ratio** detector can be computed as

$$\lambda_{L-R}(\vec{y}) = -2 \ln \left(\frac{\max_{\theta \in \mathcal{T}_0} \mathcal{L}(\vec{\theta})}{\max_{\theta \in \mathcal{T}_1} \mathcal{L}(\vec{\theta})} \right) \quad (13.275)$$

and the **generalized likelihood-ratio test (LRT)**, also known as the **Wilks test**, can be defined as

$$\lambda_{L-R}(\vec{y}) \stackrel{\mathcal{H}_1}{\underset{\mathcal{H}_0}{\gtrless}} \tau \quad (13.276)$$

where the double inequality denotes the selection of \mathcal{H}_0 or \mathcal{H}_1 depending on \vec{y} relative to τ .

This type of detector can be viewed as taking the maximum likelihood (ML) estimates of $\theta \in \mathcal{T}_0$ and $\theta \in \mathcal{T}_1$ and comparing the ratio of their likelihoods. Furthermore, **Wilks' theorem** states that the multiplication by -2 ensures that, if \mathcal{H}_0 is true, then as $n_y \rightarrow \infty$, $\lambda_{L-R}(\vec{x})$ approaches a χ^2 -distribution with degrees-of-freedom equal to the difference in dimensionality of choices within \mathcal{T}_1 and \mathcal{T}_0 . Thus, in some instances, one uses a χ^2 -squared distribution to compute the approximate τ needed for the specified α . However, Wilks' theorem requires that the “true” θ must not lie on the boundary of the sets \mathcal{T}_0 or \mathcal{T}_1 .

An important lemma of the LRT is for a “simple” null hypothesis \mathcal{H}_0 as $\theta = \vec{\theta}_0$ and a “simple” alternative hypothesis \mathcal{H}_1 as $\theta = \vec{\theta}_1$. Then, the **simple-vs.-simple likelihood-ratio** detector can be computed as

$$\Lambda(\vec{y}) = \frac{\mathcal{L}(\vec{\theta}_0)}{\mathcal{L}(\vec{\theta}_1)} \quad (13.277)$$

and the simple-vs.-simple **likelihood-ratio test (LRT)** can be defined as

$$\Lambda(\vec{y}) \stackrel{\mathcal{H}_1}{\gtrless}_{\mathcal{H}_0} \tau \quad (13.278)$$

and reject with probability q if $\Lambda(\vec{y}) = \tau$. The values τ and q are chosen to obtain some specified false alarm probability via

$$P_{FA} = \alpha = q * \Pr(\Lambda(\vec{y}) = \tau | \mathcal{H}_0) + \Pr(\Lambda(\vec{y}) < \tau | \mathcal{H}_0) \quad (13.279)$$

Furthermore, denoting the range where test decides \mathcal{H}_1 as

$$R(\tau) = \{\vec{y} : f_Y(y | \mathcal{H}_1) > \tau f_Y(y | \mathcal{H}_0)\} \quad (13.280)$$

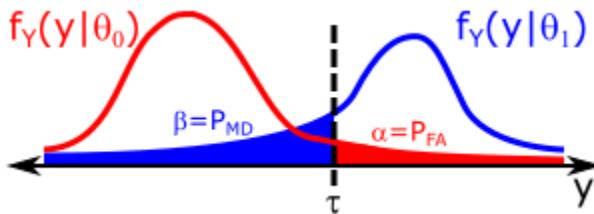
one has for the probability of false alarm

$$P_{FA} = \int_{R(\tau)} f_Y(y | \mathcal{H}_0) dy \quad (13.281)$$

and the probability of missed detection

$$P_D = \int_{R(\tau)} f_Y(y | \mathcal{H}_0) dy \quad (13.282)$$

As an example of such a test, consider two univariate PDF models and a single sample, y .



where

$$P_{FA} = \int_{\tau}^{\infty} f_Y(y|\theta_0) dy \quad (13.283)$$

and

$$P_{MD} = \int_{-\infty}^{\tau} f_Y(y|\theta_1) dy \quad (13.284)$$

Then, one can form a detector as

$$\vec{y} \underset{\mathcal{H}_0}{\gtrless} \tau \quad (13.285)$$

The **Neyman-Pearson lemma** states that for some fixed α , $\Lambda(\vec{y})$, is the **most powerful (MP) detector**, i.e. the detector D_α that maximizes the probability of correct detection, i.e. $P_D = 1 - \beta$, also known as the **power** of the detector.

$$D_\alpha^{MP}(\vec{y}) = \operatorname{argmax}_{D_\alpha \in \mathcal{D}_\alpha} P_D(D_\alpha(\vec{y})) \quad (13.286)$$

Generalized Likelihood-Ratio Tests

The **Generalized Likelihood-Ratio Test (GLRT)**

Goodness-of-Fit Tests

Kolmogorov-Smirnov statistic

Shapiro-Wilks statistic

Andersen-Darling statistic

13.7 Random Finite Sets

The underlying mathematical foundation for the general multi-sensor, multi-target problems one encountered in target tracking systems involves the mathematical foundation of point process theory which has been adapted to be “engineering-friendly” by Mahler during the 1990s and 2000s into random finite set theory which can be mechanized for multi-target problems using **finite set statistics (FISST)**, a term which Mahler coined. RFS extends Bayesian definitions for single-target filtering, by defining multi-target state spaces and measurement spaces, models, and measurement and Markov densities based on RFS definitions. A **random finite set (RFS)** \mathbf{X} is a finite-set-valued random variable which can be described by a PMF and a family of joint PDFs. The PMF characterizes the **cardinality** of \mathbf{X} , denoted by $|\mathbf{X}|$, i.e. the number of elements in the set. For a given cardinality, an appropriate PDF from the family characterizes the joint distribution of the elements of \mathbf{X} . Let $\mathcal{X} = \mathbb{R}^{n_x}$ denote the single-target state space whose elements are vectors of the form $\vec{x} = [x_1 \dots x_{n_x}]^T$ where x_1, \dots, x_{n_x} in the set of real numbers, e.g. position, velocity. Thus, the state of a multi-target system is a finite set of state vectors, i.e. $\mathbf{x} = \{\vec{x}_1, \dots, \vec{x}_N\}$ where N is the number of targets and $\vec{x}_1, \dots, \vec{x}_N$ are the individual states. The multi-target state space is the class of all finite subsets of \mathcal{X} , i.e. $\mathcal{F}(\mathcal{X})$. This allows one to define the random state set as a random finite set, \mathbf{X} .

The **RFS density** of an RFS $\mathbf{X} \in \mathcal{F}(\mathcal{X})$ is any non-negative function $f_{\mathbf{X}}(\mathbf{x})$ which is defined by the **set integral** in a region $\mathcal{S} \subseteq \mathcal{X}$ as

$$\Pr(\mathbf{x} \subseteq \mathcal{S}) = \int_{\mathcal{S}} f_{\mathbf{X}}(\mathbf{x}) \delta \mathbf{x} = \sum_{i=0}^{\infty} \frac{1}{i!} \int_{\mathcal{S}^i} f_{\mathbf{X}}(\{\vec{x}_1, \dots, \vec{x}_i\}) d(\vec{x}_1 \cdots \vec{x}_i) \quad (13.287)$$

where $f_{\mathbf{X}}(\emptyset)$ for $i = 0$. It should be noted that if $\mathcal{S} = \mathcal{X}$, then the N^{th} term for $N = 0, 1, \dots$ is the probability that there are N objects. Thus, the **cardinality distribution** of an RFS $\mathbf{X} \in \mathcal{F}(\mathcal{X})$ is defined as

$$\varrho(N) = \Pr(|\mathbf{X}| = N) \quad (13.288)$$

Notably, if \mathbb{X} and \mathbb{Y} are two independent RFSs and $\mathbb{Z} = \mathbb{X} \cup \mathbb{Y}$, then

$$f_{\mathbf{Z}}(\mathbf{z}) = \sum_{\mathbf{x} \subseteq \mathbf{z}} f_{\mathbf{X}}(\mathbf{x}) f_{\mathbf{Y}}(\mathbf{z} - \mathbf{x}) \quad (13.289)$$

which is analogous to the result that the PDF of the sum of two independent discrete random variables $Z = X + Y$ is

$$p_Z(s) = \sum_x p_X(x) p_Y(z - x) \quad (13.290)$$

To incorporate object identity, each state $\vec{x} \in \mathbb{X}$ can be augmented with a unique label $\ell \in \mathbb{L} = \{\alpha_i : i \in \mathbb{N}\}$ where \mathbb{N} denotes the set of natural numbers and α_i 's are distinct. Let $\mathcal{L} : \mathbb{X} \times \mathbb{L} \rightarrow \mathbb{L}$ be the projection $\mathcal{L}((\vec{x}, \ell)) = \ell$, then a *labeled* finite set, $\underline{\mathbf{x}}$ of $\mathbb{X} \times \mathbb{L}$, has distinct labels if and only if $\underline{\mathbf{x}}$ and its labels $\mathcal{L}(\underline{\mathbf{x}}) = \{\mathcal{L}(\vec{x}) : \vec{x} \in \underline{\mathbf{x}}\}$ have the same cardinality, i.e.

$$\Delta(\underline{\mathbf{x}}) = \delta_{|\underline{\mathbf{x}}|}(|\mathcal{L}(\underline{\mathbf{x}})|) = 1 \quad (13.291)$$

where $\Delta(\underline{\mathbf{x}})$ is the **distinct label indicator** as only distinct sets of labels will provide the correct cardinality for the N elements of $\underline{\mathbf{x}}$.

Furthermore, the **inclusion function**, a generalization of the indicator function, can be defined as

$$1_{\mathbf{y}}(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} \subseteq \mathbf{y} \\ 0, & \text{otherwise} \end{cases} \quad (13.292)$$

and one can define the mixed integral for a labeled single-target state as

$$\int f(\underline{x}) d\underline{x} = \sum_{\ell \in \mathbb{L}} \int_{\mathbb{X}} f((\vec{x}, \ell)) d\vec{x} \quad (13.293)$$

Lastly, recall that

$$h^{\mathbf{x}} = \prod_{\vec{x} \in \mathbf{x}} h(\vec{x}) \quad (13.294)$$

with $h(\emptyset) = 1$ by convention.

A **labeled random finite set (LRFS)**, $\underline{\mathbf{X}}$ with state space \mathbb{X} and discrete label space \mathbb{L} is an RFS on $\mathbb{X} \times \mathbb{L}$ such that each realization has distinct labels. Notably, the unlabeled version of a LRFS is obtained by simply discarding the labels and the cardinality distribution of an LRFS is the same as its unlabeled version. Furthermore, the set integral for the function $f : \mathcal{F}(\mathbb{X} \times \mathbb{L}) \rightarrow \mathbb{R}$ is given by

$$\int f_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) \delta \underline{\mathbf{x}} = \sum_{i=0}^{\infty} \frac{1}{i!} \sum_{(\ell_1, \dots, \ell_i) \in \mathbb{L}^i} \int_{\mathbb{X}^i} f_{\mathbf{X}}(\{(\vec{x}_1, \ell_1), \dots, (\vec{x}_i, \ell_i)\}) d(\vec{x}_1 \cdots \vec{x}_i) \quad (13.295)$$

Probability Generating Functionals of RFS

Let \mathbf{O} be an RFS of objects in some space X and given a measurable \mathcal{S} of X , let

$$\mathbf{1}_{\mathcal{S}}(\vec{x}) = \begin{cases} 1 & \text{if } \vec{x} \in \mathcal{S} \\ 0 & \text{otherwise} \end{cases} \quad (13.296)$$

Then, the **belief mass function (BMF)** can be rewritten as the expected value of the indicator function of \mathbf{O} over the subset \mathcal{S} , i.e.

$$\beta_{\mathbf{O}}(\mathcal{S}) = \prod_{i=1}^N \mathbf{1}_{\mathcal{S}}(\vec{x}_i) f_{\mathbf{O}}(\mathbf{x}) \delta \mathbf{x} = \int \mathbf{1}_{\mathcal{S}}^{\mathbf{O}} f_{\mathbf{O}}(\mathbf{x}) \delta \mathbf{x} \quad (13.297)$$

Then, by defining for any finite subset, \mathbf{x} of X and any real-valued function of \mathbf{x} ,

$$h^{\mathbf{x}} = \begin{cases} 1 & \text{if } \mathbf{x} = \emptyset \\ \prod_{i=1}^N h(\vec{x}_i) & \text{if } \mathbf{x} = \{\vec{x}_1, \dots, \vec{x}_N\} \text{ with } \vec{x}_1, \dots, \vec{x}_N \text{ distinct} \end{cases} \quad (13.298)$$

one can generalize the concept of the BMF by replacing $\mathbf{1}_{\mathcal{S}}(\vec{x})$ with any $h(\vec{x})$ such that

$$h(\vec{x}) = h_0(\vec{x}) + w_1 \delta_{\vec{w}_1}(\vec{x}) + \dots + w_{n_x} \delta_{\vec{w}_{n_x}}(\vec{x}) \quad (13.299)$$

where $0 \leq h_0(\vec{x}) \leq 1$ has no units of measurement, $\delta_w(\vec{x})$ is the Dirac delta function, $\vec{w}_1, \dots, \vec{w}_{n_x}$ are fixed distinct elements of X , and w_1, \dots, w_{n_x} have the same units of measurement as \vec{x} .

Then, one can define the **probability generating functional (PGFL)** of \mathbf{X} as

$$G_{\mathbf{O}}[h] = \int h^{\mathbf{x}} f_{\mathbf{O}}(\mathbf{x}) \delta \mathbf{x} \quad (13.300)$$

and

$$\beta_{\mathbf{O}}(\mathcal{S}) = G_{\mathbf{O}}[\mathbf{1}_{\mathcal{S}}] \quad (13.301)$$

The PGFL is well-defined and finite valued because $f_{\mathbf{O}}(\{\vec{x}_1, \dots, \vec{x}_i, \dots, \vec{x}_j, \dots, \vec{x}_N\}) = 0$ whenever $\vec{x}_i = \vec{x}_j$ for $i \neq j$, so undefined products of $\delta_u(\vec{y})^2$ do not occur. The intuitive meaning of the PGFL is as follows. Let X be a single-target state space, $\mathbf{O} = \mathbf{x}$, a random finite subset of X , and $0 \leq h(\vec{x}) \leq 1$, so that $h(\vec{x})$ can be interpreted as the probability of detection for the surveillance volume (SV) or field-of-view (FOV) of a sensor. Then, it can be shown that $G_{\mathbf{x}}[h]$ is the probability that \mathbf{x} is contained in the surveillance volume. Furthermore, as $h(\vec{x})$ is a fuzzy membership function on \mathbb{R}^{n_x} , $G_{\mathbf{x}}[h]$ is a generalization of the BMF from hard sets X to fuzzy subsets h .

Importantly,

$$G_{\mathbf{Z}}[h] = G_{\mathbf{X}}[h] G_{\mathbf{Y}}[h] \quad (13.302)$$

which is analogous to the result that the PGF of the sum of two independent random variables $Z = X + Y$ is

$$G_Z(s) = G_X(s) G_Y(s) \quad (13.303)$$

Moments of Random Finite Sets

For an RFS \mathbf{X} with RFS density $f_{\mathbf{X}}(\vec{x})$, the **RFS moment density** is defined as

$$D_{\mathbf{X}}(\mathbf{x}) = \int f_{\mathbf{X}}(\mathbf{x} \cup \mathbf{w}) \delta \mathbf{w} \quad (13.304)$$

where $D_{\mathbf{X}}(\emptyset) = 1$ and the set integral is well-defined. An intuitive interpretation of the moment density is for any $\mathbf{x} = \{\vec{x}_1, \dots, \vec{x}_N\}$, $D_{\mathbf{X}}(\mathbf{x})$ is the probability density that N of the targets in \mathbf{X} have states $\vec{x}_1, \dots, \vec{x}_N$. The N^{th} -order **RFS moment density** is defined as

$$D_{\mathbf{X}}(\{\vec{x}_1, \dots, \vec{x}_N\}) \quad (13.305)$$

Furthermore, the **first-order RFS moment density** is defined as

$$\nu(\vec{x}) = D_{\mathbf{X}}(\{\vec{x}\}) = \int f_{\mathbf{X}}(\{\vec{x}\} \cup \mathbf{w}) \delta \mathbf{w} = \int \delta_{\mathbf{x}}(\vec{x}) f_{\mathbf{X}}(\vec{x}) \delta \mathbf{x} \quad (13.306)$$

which holds by setting $h(\vec{x}) = \delta_{\mathbf{x}}(\vec{x})$. This is also known as the **probability hypothesis density (PHD)** or **intensity function** of \mathbf{X} , so named because one can show that for any subset $\mathcal{S} \subseteq \mathcal{X}$

$$\hat{N}(\mathcal{S}) = \mathbb{E}[|\mathbf{X} \cap \mathcal{S}|] = \int_{\mathcal{S}} \nu(\vec{x}) d\vec{x} \quad (13.307)$$

Intuitively, the local maxima of the PHD are points in \mathcal{X} with the highest local concentration of expected number of elements. Intuitively, one can use $\hat{N} = \mathbb{E}[|\mathbf{X}|]$ as the estimated number of targets and the locations of the \hat{N} highest local maxima of the PHD as the estimated target states.

Fundamental RFS Distributions

A **IID cluster RFS**, \mathbf{X} , has elements IID according to a PDF $f_{\mathbf{X}}(\vec{x})$ and is completely characterized by ϱ and f . Its RFS density is defined as

$$f_{\mathbf{X}}(\{\vec{x}_1, \dots, \vec{x}_N\}) = N! \varrho(N) \prod_{i=1}^N f_{\vec{X}}(\vec{x}_i) \quad (13.308)$$

with $f_{\mathbf{X}}(\emptyset) = \varrho(0)$. A **Poisson RFS**, also known as a **Poisson Point Process (PPP) RFS** is a special case of an IID cluster RFS with a Poisson cardinality distribution, i.e.,

$$\varrho(N) = \Pr(|\mathbf{x}| = N) = e^{-\lambda} \frac{\lambda^N}{N!} \quad (13.309)$$

where mean value, $\hat{N} = \lambda$, also known as the Poisson rate. Thus, the Poisson RFS density is given by

$$f_{\mathbf{X}}(\mathbf{x}) = \exp \left(- \int \lambda(x) dx \right) = e^{-\lambda} \lambda^N \prod_{j=1}^N f_{\vec{X}}(\vec{x}_j) \quad (13.310)$$

where, by definition of the set integral, one has

$$\begin{aligned}
 \int f_{\mathbf{X}}\{\vec{x}\} \delta \mathbf{x} &= \sum_{i=0}^{\infty} \frac{1}{i!} \int f_{\mathbf{X}}(\{\vec{x}_1, \dots, \vec{x}_i\}) d(\vec{x}_1 \cdots \vec{x}_i) \\
 &= \sum_{i=0}^{\infty} \frac{1}{i!} \int_{S^i} e^{-\lambda} \lambda^i \prod_{j=1}^i f_{\vec{X}}(\vec{x}_j) d(\vec{x}_1 \cdots \vec{x}_i) \\
 &= e^{-\lambda} \sum_{i=0}^{\infty} \frac{\lambda^i}{i!} \\
 &= e^{-\lambda} e^{\lambda} = 1
 \end{aligned} \tag{13.311}$$

Furthermore, by definition of the PHD $\nu(\vec{x})$ as

$$\nu(\vec{x}) = \int f_{\mathbf{X}}(\vec{x} \cup \mathbf{w}) \delta \mathbf{w} \tag{13.312}$$

which, by definition of the set integral, one has

$$\begin{aligned}
 \nu(\vec{x}) &= \sum_{i=0}^{\infty} \frac{1}{i!} \int_{S^i} f_{\mathbf{X}}(\{\vec{x}, \vec{w}, \dots, \vec{w}_i\}) d\vec{x} \\
 &= e^{-\lambda} f_{\vec{X}}(\vec{x}) \sum_{i=0}^{\infty} \lambda^{i+1} i! \int_{S^i} f_{\mathbf{X}}(\{\vec{w}, \dots, \vec{w}_i\}) d(\vec{w}_1 \cdots \vec{w}_i) \\
 &= \lambda f_{\vec{X}}(\vec{x})
 \end{aligned} \tag{13.313}$$

Thus, a Poisson RFS with cardinality parameter, λ , and single-target PDF, $f_{\vec{X}}(\vec{x})$, as one has

$$f_{\vec{X}}(\vec{x}) = \frac{\nu(\vec{x})}{\lambda} \tag{13.314}$$

$$f_{\mathbf{X}}(\mathbf{x}) = e^{-\lambda} \prod_{j=1}^N \frac{\nu(\vec{x}_j)}{\lambda} \tag{13.315}$$

and

$$\hat{N} = \lambda = \int_{\mathcal{X}} \nu(\vec{x}) d\vec{x} \tag{13.316}$$

and the PGFL is given by

$$G_{\mathbf{X}}[h] = \exp \tag{13.317}$$

A Bernoulli RFS, \mathbf{X} has probability $1 - r$ of being empty, and probability r of being a singleton whose element is distributed according to a PDF $f_{\vec{X}}(\vec{x})$. The RFS density of the Bernoulli RFS is given by

$$f_{\mathbf{X}}(\mathbf{x}) = \begin{cases} 1 - r & \mathbf{x} = \emptyset \\ r f_{\vec{X}}(\vec{x}) & \mathbf{x} = \{\vec{x}\} \end{cases} \tag{13.318}$$

Furthermore, a **multi-Bernoulli RFS** is a union of independent Bernoulli RFSSs. The multi-Bernoulli RFS density is given by

$$f_{\mathbf{X}}(\mathbf{x}) = \begin{cases} 1 - r & \mathbf{x} = \emptyset \\ rf_{\vec{X}}(\vec{x}) & \mathbf{x} = \{\vec{x}\} \end{cases} \quad (13.319)$$

and the PGFL is given by

$$G_{\mathbf{X}}[h] = \quad (13.320)$$

Lastly, a **multi-Bernoulli mixture RFS** is a weighted sum of independent multi-Bernoulli RFSSs. The multi-Bernoulli mixture RFS density is given by

$$f_{\mathbf{X}}(\mathbf{x}) = \begin{cases} 1 - r & \mathbf{x} = \emptyset \\ rf_{\vec{X}}(\vec{x}) & \mathbf{x} = \{\vec{x}\} \end{cases} \quad (13.321)$$

and the PGFL is given by

$$G_{\mathbf{X}}[h] = \quad (13.322)$$

A **labeled IID cluster RFS**, $\underline{\mathbf{X}}$, has an RFS density given by

$$f_{\underline{\mathbf{X}}}(\{(\vec{x}_1, \ell_1), \dots, (\vec{x}_N, \ell_N)\}) = \delta_{\{\alpha_1, \dots, \alpha_N\}}(\{\ell_1, \dots, \ell_N\}) \varrho(N) \prod_{i=1}^N \frac{\nu(\vec{x}_i)}{\langle v, 1 \rangle} \quad (13.323)$$

where $\mathbb{L}(N) = \{\alpha_i \in \mathbb{L}\}_{i=1}^N$, ν is the PHD of the unlabeled states, and $\varrho(N)$ is the cardinality distribution. For a **labeled Poisson RFS**, one has

$$\varrho(N) = e^{-\lambda} \frac{\lambda^N}{N!} \quad (13.324)$$

where $\lambda = \langle v, 1 \rangle$.

The **labeled multi-Bernoulli (LMB) RFS**, $\underline{\mathbf{X}}$, has a finite parameter set $(r^{(\zeta)}, f_{\vec{X}}^{(\zeta)}) : \zeta \in \Psi$ where the labels in \mathbb{L} are “assigned” to non-empty Bernoulli components, i.e. if $(r^{(\zeta)}, f_{\vec{X}}^{(\zeta)})$ is non-empty, then the label is given by $\alpha(\zeta)$ where $\alpha : \Psi \rightarrow \mathbb{L}$ is a 1-1 mapping. Then, the multi-target FISST density is given by

$$f_{\underline{\mathbf{X}}}(\{(\vec{x}_1, \ell_1), \dots, (\vec{x}_N, \ell_N)\}) = \delta_N(|\{\ell_1, \dots, \ell_N\}|) \prod_{\zeta \in \Psi} (1 - r^{(\zeta)}) \prod_{j=1}^N \frac{1_{\alpha(\Psi)}(\ell_j) r^{(\alpha^{-1}(\ell_j))} f_{\vec{X}}^{(\alpha^{-1}(\ell_j))}(\vec{x}_j)}{1 - r^{(\alpha^{-1}(\ell_j))}} \quad (13.325)$$

which can be written alternatively as

$$f_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) = \Delta(\underline{\mathbf{x}}) 1_{\alpha(\Psi)}(\mathcal{L}(\underline{\mathbf{x}})) [\Phi(\vec{x}; \cdot)]^{\Psi} \quad (13.326)$$

where

$$\Phi(\vec{x}; \zeta) = \begin{cases} 1 - r^{(\zeta)}, & \text{if } \alpha(\zeta) \notin \mathcal{L}(\underline{\mathbf{x}}) \\ r^{(\zeta)} f_{\vec{X}}^{(\zeta)}(\vec{x}) & \text{if } (\vec{x}, \alpha(\zeta)) \in \underline{\mathbf{x}} \end{cases} \quad (13.327)$$

With these two LRFS in mind, the **generalized labeled multi-Bernoulli (GLMB) RFS** can be defined as

$$f_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) = \Delta(\underline{\mathbf{x}}) \sum_{\xi \in \Xi} w^{(\xi)}(\mathcal{L}(\underline{\mathbf{x}})) [f_{\vec{X}}^{(\xi)}]^{\underline{\mathbf{x}}} \quad (13.328)$$

where Ξ is a discrete index set, $w^{(\xi)}(L)$ satisfies

$$\sum_{L \subseteq \mathbb{L}} \sum_{\xi \in \Xi} w^{(\xi)}(L) = 1 \quad (13.329)$$

and $f_{\underline{\mathbf{x}}}^{(\xi)}$ satisfies

$$\int f_{\underline{\mathbf{x}}}^{(\xi)}(\vec{x}, \ell) d\vec{x} = 1 \quad (13.330)$$

A GLMB can be interpreted as a mixture of multi-target exponentials where each term consists of a weight $w^{(\xi)}(\mathcal{L}(\underline{\mathbf{x}}))$ that only depends on the labels of the multi-target state, and a multi-target exponential $[f_{\underline{\mathbf{x}}}^{(\xi)}]^{\underline{\mathbf{x}}}$ that depends on the entire multi-object state. The points of a GLMB RFS are statistically dependent.

Notably, the PHD of the unlabeled version of the GLMB RFS is

$$\nu(\vec{x}) = \sum_{\xi \in \Xi} \sum_{\ell \in \mathbb{L}} f_{\underline{\mathbf{x}}}^{(\xi)}(\vec{x}, \ell) \sum_{L \subseteq \mathbb{L}} 1_L(\ell) w^{(\xi)}(L) \quad (13.331)$$

and the cardinality distribution of a GLMB RFS is given by

$$\varrho(N) = \sum_{L \in \mathcal{F}_N(\mathbb{L})} \sum_{\xi \in \Xi} w^{(\xi)}(L) \quad (13.332)$$

where $\mathcal{F}_N(\mathbb{L})$ denotes the subsets of \mathbb{L} with exactly N elements.

Thus, the GLMB RFS is flexible enough to approximate any labeled RFS density with matching PHD and cardinality distribution, including the labeled Poisson RFS which sets

$$\begin{aligned} w^{(\xi)}(L) &= \delta_{\mathbb{L}(|L|)}(L) \varrho(|L|) \\ f_{\underline{\mathbf{x}}}^{(\xi)}(\vec{x}, \ell) &= \frac{\nu(\vec{x})}{\langle \nu, 1 \rangle} \end{aligned} \quad (13.333)$$

and the LMB RFS which sets

$$\begin{aligned} w^{(\xi)}(L) &= \prod_{i \in \mathbb{L}} (1 - r^{(i)}) \prod_{\ell \in L} \frac{1_{\mathbb{L}}(\ell) r^{(\ell)}}{1 - r^{(\ell)}} \\ f_{\underline{\mathbf{x}}}^{(\xi)}(\vec{x}, \ell) &= f_{\underline{\mathbf{x}}}^{(\ell)}(\vec{x}, \ell) \end{aligned} \quad (13.334)$$

both of which have an index space Ξ with one element, thus, not requiring the (ξ) superscript.

Optimal Parameter Estimation Theory

14.1 Introduction to Optimal Parameter Estimation

Decision theory is the determination of the optimal decision given constraints and assumptions on a decision problem. **Parameter estimation** is one special case of a decision problem and can be stated as deciding which parameter(s) one should select and the general **parameter estimation problem** can be considered as follows. Let \vec{y} be samples of a random vector \vec{Y} which depend on some *uncertain* parameter vector, $\vec{\beta}$. As such, this probabilistic relationship can be described using a conditional PDF, $f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta})$, or as a **likelihood function** $\mathcal{L}(\vec{\beta}|\vec{y})$ which are functionally equivalent, but view the relationships from different viewpoints. Using this information, one generally forms some mathematical law called the **parameter estimator** of $\vec{\beta}$ based on \vec{y} and is typically denoted by $\hat{\vec{\beta}}(\vec{y})$. An **optimal parameter estimator** is used when one forms the mathematical law through an optimization with respect to a chosen attribute or statistic of $f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta})$.

Maximum Likelihood Estimator

An example of an optimal parameter estimator is the **maximum likelihood estimator (MLE)** which can be stated as

$$\hat{\vec{\beta}}_{MLE} = \underset{\vec{\beta}}{\operatorname{argmax}} \mathcal{L}(\vec{\beta}|\vec{y}) \quad (14.1)$$

which is often computed using the log-likelihood as

$$\hat{\vec{\beta}}_{MLE} = \underset{\vec{\beta}}{\operatorname{argmin}} -\ln \mathcal{L}(\vec{\beta}|\vec{y}) = \underset{\vec{\beta}}{\operatorname{argmin}} \quad (14.2)$$

The MLE exists if the Jacobian of the log-likelihood is the zero row vector. For some probability models, one can directly compute $\hat{\vec{\beta}}_{MLE}$ through solving the gradient equation

$$\nabla \ln \mathcal{L}(\hat{\vec{\beta}}_i|\vec{y}) = \frac{\partial \ln \mathcal{L}(\vec{\beta}|\vec{y})}{\partial \vec{\beta}} = 0 \quad (14.3)$$

where the gradient is also known as the **score**, $\vec{s}_i(\hat{\beta})$, for iteration i . However, in general, no closed-form solution to the MLE problem is known and can only be found via numerical optimization methods.

The most common optimization methods for MLE are iterative procedures of the form:

$$\hat{\beta}_{i+1} = \hat{\beta}_i + \eta_i \vec{d}_i(\hat{\beta}) \quad (14.4)$$

where η_i is the step length, i.e. **learning rate** and \vec{d}_i is the **descent direction**. For choosing these values, one can use the Gradient Descent method where η_i is chosen small enough for convergence and $\vec{d}_i = \nabla \ln \mathcal{L}(\hat{\beta}_i | \vec{y})$. One can also use the Newton-Raphson method where $\eta_i = 1$ and $\vec{d}_i = -H_i^{-1}(\hat{\beta}) \vec{s}_i(\hat{\beta})$. Then, $H_i^{-1}(\hat{\beta})$: inverse of Hessian matrix of log-likelihood function, i.e.,

$$H(\hat{\beta}) = -\frac{\partial^2 \ln \mathcal{L}(\hat{\beta} | \vec{y})}{\partial \hat{\beta}^2} \quad (14.5)$$

Quasi-Newton methods which attempt to approximate the Hessian matrix are the Davidson-Fletcher-Powell (DFP) formula, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm, and the Fisher's Scoring which replaces the explicit Hessian with **Fisher information matrix**, i.e., the covariance of the score as

$$\mathcal{I}(\hat{\beta}) = \mathbb{E}\left[\left(\frac{\partial \ln \mathcal{L}(\hat{\beta} | \vec{y})}{\partial \hat{\beta}}\right)\left(\frac{\partial \ln \mathcal{L}(\hat{\beta} | \vec{y})}{\partial \hat{\beta}}\right)^T\right] \quad (14.6)$$

For a single observed sample, \vec{y} , the MLE has no guaranteed performance. However, the MLE has some nice properties as the number of repeated samples approaches infinity, i.e. for repeated \vec{y}_k with $k = 1, 2, 3, \dots$. Namely, if each sample is **independent and identically distributed (IID)**, then the combined likelihood can be written simply as a product of the conditional PDFs or likelihoods as

$$\hat{\beta}_{MLE} = \underset{\hat{\beta}}{\operatorname{argmin}} \prod_{k=1}^N \mathcal{L}(\hat{\beta} | \vec{y}_k) \quad (14.7)$$

or, in terms of the log-likelihood, as

$$\hat{\beta}_{MLE} = \underset{\hat{\beta}}{\operatorname{argmin}} - \sum_{k=1}^N \ln \mathcal{L}(\hat{\beta} | \vec{y}_k) \quad (14.8)$$

It can be shown that as $N \rightarrow \infty$, $\hat{\beta}_{MLE}$ converges in probability to $\vec{\beta}$, a property known as **estimator consistency**.

Method of Moments

An alternative to the MLE for IID samples which characterize a conditional PDF, $f_Y(y | \vec{\beta})$, is the **method of moments (MOM) estimator** which can be used for estimating the k unknown parameters of the distribution, $\vec{\beta}$.

In basic MOM, one can form the k moments of Y as

$$\begin{bmatrix} m_1 \\ \vdots \\ m_k \end{bmatrix} = \begin{bmatrix} \mathbb{E}[Y] \\ \vdots \\ \mathbb{E}[Y^k] \end{bmatrix} = \begin{bmatrix} g_1(\beta_1, \dots, \beta_k) \\ \vdots \\ g_k(\beta_1, \dots, \beta_k) \end{bmatrix} \quad (14.9)$$

Next, suppose that N samples are drawn of Y , denoted as y_i for $i = 1, \dots, N$, then one can form the empirical moments

$$\begin{bmatrix} \hat{m}_1 \\ \vdots \\ \hat{m}_k \end{bmatrix} = \begin{bmatrix} \frac{1}{N} \sum_{i=1}^N y_i \\ \vdots \\ \frac{1}{N} \sum_{i=1}^N y_i \end{bmatrix} \quad (14.10)$$

Finally, the basic MOM estimator solves the equations

$$\begin{bmatrix} \hat{m}_1 \\ \vdots \\ \hat{m}_k \end{bmatrix} = \begin{bmatrix} g_1(\hat{\beta}_1, \dots, \hat{\beta}_k) \\ \vdots \\ g_k(\hat{\beta}_1, \dots, \hat{\beta}_k) \end{bmatrix} \quad (14.11)$$

An alternative to the basic moments of distributions are **probability weighted moments (PWM)** defined for three parameters (p, r, s) as

$$\vec{M}(p, r, s) = \mathbb{E}[Y^p F_Y(y)^r (1 - F_Y(y))^s] \quad (14.12)$$

which can be used to yield a MOM estimator. A special case of PWMs are the **L-moments** where the r^{th} L-moment, λ_r , can be defined as

$$\lambda_r = r^{-1} \sum_{j=0}^{r-1} (-1)^j \frac{(r-1)!}{(r-j-1)! j!} \mathbb{E}[Y_{r-j:r}] \quad (14.13)$$

where $Y_{k:N}$ denotes the k^{th} **order statistic**, i.e. the k^{th} smallest value, in the N IID samples. Furthermore, the empirical L-moments can be defined as

$$\hat{\lambda}_r = r^{-1} \frac{(N-r)! r!}{N!} \sum_{y_1 < \dots < y_j < \dots < y_r} (-1)^{r-j} \frac{(r-1)!}{(r-j-1)! j!} y_j \quad (14.14)$$

where one has summed over the r -element subset of the samples $\{y_1 < \dots < y_j < \dots < y_r\}$, thereby averaging by dividing by the binomial coefficient.

In Generalized MOM, one must be able to form a vector-valued moment conditions, $g(\vec{\beta})$ for the moment functions, $\vec{m} \in \mathbb{R}^k$, of the conditional PDF, i.e.

$$\vec{m}(\vec{\beta}) = \mathbb{E}[g(\vec{y}, \vec{\beta})] = 0 \quad (14.15)$$

Next, let the empirical moments, $\hat{m}(\vec{\beta})$, be defined as

$$\hat{m}(\vec{\beta}) = \frac{1}{N} \sum_{i=1}^N g(\vec{y}_i, \vec{\beta}) \quad (14.16)$$

Then, the MOM estimator can be stated as

$$\hat{\vec{\beta}}_{MOM} = \underset{\hat{\vec{\beta}}}{\operatorname{argmin}} \left(\frac{1}{N} \sum_{k=1}^N g(\vec{y}, \vec{\theta}) \right)^T \left(\frac{1}{N} \sum_{k=1}^N g(\vec{y}, \hat{\vec{\beta}}) \right) \quad (14.17)$$

where this norm minimization attempts to look for a θ which makes \hat{m} as close to zero as possible which should be the case by the law of large numbers (for N).

Estimator Error Optimality

One important characteristic of any parameter estimator is the **estimator error**, i.e. $\hat{\vec{\beta}} - \vec{\beta}$, which is a random vector that can generally have *positive or negative* values. Thus, a common statistic for any estimator considers the expectation of the square of the error, also known as the **mean square error (MSE)** of the estimator, defined as

$$\text{MSE}(\hat{\vec{\beta}}) = \mathbb{E} \left[(\hat{\vec{\beta}} - \vec{\beta})^T (\hat{\vec{\beta}} - \vec{\beta}) \right] \quad (14.18)$$

Thus, an optimal parameter estimator may be formed as

$$\hat{\vec{\beta}}_{MMSE} = \underset{\hat{\vec{\beta}}}{\operatorname{argmin}} \mathbb{E} \left[(\hat{\vec{\beta}} - \vec{\beta})^T (\hat{\vec{\beta}} - \vec{\beta}) \right] \quad (14.19)$$

which is known as the **minimum MSE (MMSE) estimator**. For the MLE, it can be shown that as $N \rightarrow \infty$, the MLE becomes the MMSE.

Another statistic of an estimator is the **estimator mean**, i.e. $\mathbb{E}[\hat{\vec{\beta}}]$, which may or may not match the actual parameter $\vec{\beta}$. Thus, one often may consider the **estimator bias**, $\vec{b}(\hat{\vec{\beta}})$ defined as

$$\vec{b}(\hat{\vec{\beta}}) = \mathbb{E}[\hat{\vec{\beta}}] - \vec{\beta} \quad (14.20)$$

Thus, if for some $\hat{\vec{\beta}}$, $\mathbb{E}[\hat{\vec{\beta}}] = \vec{\beta}$, then $\hat{\vec{\beta}}$ is **unbiased**. A third statistic of an estimator is the **estimator variance**, i.e.

$$\text{Var}(\hat{\vec{\beta}}) = \mathbb{E} \left[(\hat{\vec{\beta}} - \mathbb{E}[\hat{\vec{\beta}}])^T (\hat{\vec{\beta}} - \mathbb{E}[\hat{\vec{\beta}}]) \right] \quad (14.21)$$

With these two characteristics in mind, an optimal parameter estimator may be formed as

$$\hat{\vec{\beta}}_{MVUE} = \underset{\text{unbiased } \vec{\beta}}{\operatorname{argmin}} \mathbb{E} \left[(\hat{\vec{\beta}} - \mathbb{E}[\hat{\vec{\beta}}])^T (\hat{\vec{\beta}} - \mathbb{E}[\hat{\vec{\beta}}]) \right] \quad (14.22)$$

which is known as the **minimum variance unbiased estimator (MVUE)**.

Next, note that the MSE of $\hat{\vec{\beta}}$ can be written as

$$\text{MSE}(\hat{\vec{\beta}}) = \mathbb{E} \left[\hat{\vec{\beta}}^T \hat{\vec{\beta}} - \hat{\vec{\beta}}^T \vec{\beta} - \vec{\beta}^T \hat{\vec{\beta}} + \vec{\beta}^T \vec{\beta} \right] \quad (14.23)$$

$$\text{MSE}(\hat{\vec{\beta}}) = \mathbb{E}[\hat{\vec{\beta}}^T \hat{\vec{\beta}}] - \mathbb{E}[2\hat{\vec{\beta}}^T \vec{\beta}] + \mathbb{E}[\vec{\beta}^T \vec{\beta}] \quad (14.24)$$

or

$$\text{MSE}(\hat{\beta}) = \mathbb{E}[\hat{\beta}^T \hat{\beta}] - 2\mathbb{E}[\hat{\beta}^T] \vec{\beta} - \vec{\beta}^T \vec{\beta} \quad (14.25)$$

Also, note that the square of the bias can be written as

$$\vec{b}(\hat{\beta})^2 = \mathbb{E}[\hat{\beta}] \mathbb{E}[\hat{\beta}]^T - 2\mathbb{E}[\hat{\beta}] \vec{\beta} - \vec{\beta}^T \vec{\beta} \quad (14.26)$$

and that the variance can be written as

$$\text{Var}(\hat{\beta}) = \mathbb{E}\left[\hat{\beta}^T \hat{\beta} - 2\mathbb{E}[\hat{\beta}]^T \hat{\beta} - \mathbb{E}[\hat{\beta}]^T \mathbb{E}[\hat{\beta}]\right] \quad (14.27)$$

or

$$\text{Var}(\hat{\beta}) = \mathbb{E}\left[\hat{\beta}^T \hat{\beta}\right] - 2\mathbb{E}[\hat{\beta}]^T \mathbb{E}[\hat{\beta}] - \mathbb{E}[\hat{\beta}]^T \mathbb{E}[\hat{\beta}] \quad (14.28)$$

Thus, by comparing the MSE, variance, and bias, one can show that

$$\text{MSE}(\hat{\beta}) = \text{Var}(\hat{\beta}) + \vec{b}(\hat{\beta})^2 \quad (14.29)$$

which demonstrates that the MVUE is equivalent to the MMSE among unbiased estimators.

Lastly, the **Cramer-Rao Bound (CRB)** states that the lower bound on the covariance of any *unbiased* estimator of $\vec{\beta}$ can be shown to be the inverse of Fisher information matrix, i.e.

$$\text{Cov}(\hat{\beta}) \geq \mathcal{I}^{-1}(\vec{\beta}) \quad (14.30)$$

where an estimator which achieves the CRB is known as an **efficient estimator**. Furthermore, under certain conditions, it can also be shown that the MLE is asymptotically efficient and its estimator error converges in distribution to a normal distribution, i.e.

$$\sqrt{N}(\hat{\beta}) - \vec{\beta} \rightarrow \mathcal{N}(0, \mathcal{I}^{-1}) \quad (14.31)$$

Linear Parameter Estimators

In many cases, one desires to use a **linear parameter estimator**, i.e.

$$\hat{\beta} = L \vec{y} \quad (14.32)$$

where L is the **estimator gain matrix**. Then, for a MVUE that is also linear, one can form the optimization

$$\hat{\beta}_{BLUE} = \underset{\text{unbiased, linear } \vec{\beta}}{\operatorname{argmin}} \mathbb{E}\left[(\hat{\beta} - \mathbb{E}[\hat{\beta}])^T (\hat{\beta} - \mathbb{E}[\hat{\beta}])\right] \quad (14.33)$$

which is known as the **best linear unbiased estimator (BLUE)**.

As an example, consider a linear measurement, observation, or regression model, i.e.

$$\vec{y} = X \vec{\beta} + \vec{\epsilon} \quad (14.34)$$

where X is known as the **measurement matrix**, also known as the **observation matrix**, **regression matrix**, or the **design matrix**, and $\vec{\epsilon}$ is zero-mean **measurement error**, also known as the **observation error** or **regression error**, i.e., $\mathbb{E}[\vec{\epsilon}] = 0$, which is an arbitrary assumption since one can simply form a new linear regression model, \vec{y}' , with zero-mean regression error, $\vec{\epsilon}'$, as

$$\vec{y}' = \vec{y} - \mathbb{E}[\vec{\epsilon}] = X\vec{\beta} + \vec{\epsilon} - \mathbb{E}[\vec{\epsilon}] = X\vec{\beta} + \vec{\epsilon}' \quad (14.35)$$

Furthermore, assume that the covariance of $\vec{\epsilon}$ is

$$\mathbb{E}[\vec{\epsilon}\vec{\epsilon}^T] = \sigma^2 I \quad (14.36)$$

With this model, the mean of \vec{y} can be shown to be

$$\mathbb{E}[\vec{y}] = \mathbb{E}[X\vec{\beta} + \vec{\epsilon}] \quad (14.37)$$

$$\mathbb{E}[\vec{y}] = \mathbb{E}[X\vec{\beta}] + \mathbb{E}[\vec{\epsilon}] \quad (14.38)$$

$$\mathbb{E}[\vec{y}] = X\vec{\beta} \quad (14.39)$$

which demonstrates that this linear model is modeling the mean of samples as a linear combination of the unknown parameters.

The covariance of \vec{y} can be shown to be

$$\text{Cov}(\vec{y}) = \mathbb{E}[(\vec{y} - \mathbb{E}[\vec{y}])(\vec{y} - \mathbb{E}[\vec{y}])^T] \quad (14.40)$$

$$\text{Cov}(\vec{y}) = \mathbb{E}[(X\vec{\beta} + \vec{\epsilon} - X\vec{\beta})(X\vec{\beta} + \vec{\epsilon} - X\vec{\beta})^T] \quad (14.41)$$

$$\text{Cov}(\vec{y}) = \mathbb{E}[\vec{\epsilon}\vec{\epsilon}^T] \quad (14.42)$$

which assumes that the samples have variance modeled by the additive error random vector.

If one assumes that

$$\text{Cov}(\vec{y}) = \sigma^2 I \quad (14.43)$$

Then, it can be shown that the BLUE is given by

$$\hat{\vec{\beta}}_{BLUE} = (X^T X)^{-1} X^T \vec{y} \quad (14.44)$$

with covariance

$$\text{Cov}(\hat{\vec{\beta}}_{BLUE}) = \text{Cov}((X^T X)^{-1} X^T \vec{y}) \quad (14.45)$$

$$\text{Cov}(\hat{\vec{\beta}}_{BLUE}) = (X^T X)^{-1} X^T \text{Cov}(\vec{y}) X (X^T X)^{-1} \quad (14.46)$$

$$\text{Cov}(\hat{\vec{\beta}}_{BLUE}) = (X^T X)^{-1} X^T (\sigma^2 I) X (X^T X)^{-1} \quad (14.47)$$

$$\text{Cov}(\hat{\vec{\beta}}_{BLUE}) = \sigma^2 (X^T X)^{-1} \quad (14.48)$$

As a proof, assume a different linear unbiased estimator exists with minimum variance, i.e.

$$\tilde{\vec{\beta}} = \left((X^T X)^{-1} X^T + D \right) \vec{y} \quad (14.49)$$

where D is some non-zero matrix. Then, computing expectation of this estimator allows one to find the condition for which $\tilde{\beta}$ is unbiased, i.e.

$$\mathbb{E} \left[\tilde{\beta} \right] = \mathbb{E} \left[\left((X^T X)^{-1} X^T + D \right) \vec{y} \right] \quad (14.50)$$

Substituting for \vec{y}

$$\mathbb{E} \left[\tilde{\beta} \right] = \mathbb{E} \left[\left((X^T X)^{-1} X^T + D \right) (X \beta + \vec{\epsilon}) \right] \quad (14.51)$$

Distributing the expectation for the random vector term

$$\mathbb{E} \left[\tilde{\beta} \right] = \left((X^T X)^{-1} X^T + D \right) X \beta + \mathbb{E} \left[\left((X^T X)^{-1} X^T + D \right) \vec{\epsilon} \right] \quad (14.52)$$

which can be simplified since $\vec{\epsilon}$ is zero-mean as

$$\mathbb{E} \left[\tilde{\beta} \right] = \left((X^T X)^{-1} X^T + D \right) X \beta \quad (14.53)$$

and rearranging

$$\mathbb{E} \left[\tilde{\beta} \right] = (X^T X)^{-1} X^T X \beta + D X \beta \quad (14.54)$$

or

$$\mathbb{E} \left[\tilde{\beta} \right] = (I + D X) \beta \quad (14.55)$$

which is unbiased only if $D X = 0$.

Then, one can inspect the covariance to see if this new estimator can potentially have a lower variance

$$\text{Cov} \left(\tilde{\beta} \right) = \text{Cov} (L \vec{y}) \quad (14.56)$$

$$\text{Cov} \left(\tilde{\beta} \right) = L \text{Cov} (\vec{y}) L^T \quad (14.57)$$

$$\text{Cov} \left(\tilde{\beta} \right) = \sigma^2 L L^T \quad (14.58)$$

Substituting the potential linear estimator for L

$$\text{Cov} \left(\tilde{\beta} \right) = \sigma^2 \left((X^T X)^{-1} X^T + D \right) \left(X (X^T X)^{-1} + D^T \right) \quad (14.59)$$

which can be distributed as

$$\text{Cov} \left(\tilde{\beta} \right) = \sigma^2 \left((X^T X)^{-1} X^T X (X^T X)^{-1} + (X^T X)^{-1} X^T D^T + D X (X^T X)^{-1} + D D^T \right) \quad (14.60)$$

$$\text{Cov} \left(\tilde{\beta} \right) = \sigma^2 \left((X^T X)^{-1} + (X^T X)^{-1} (D X)^T + D X (X^T X)^{-1} + D D^T \right) \quad (14.61)$$

and simplifies to the following since $D X = 0$ must be zero

$$\text{Cov} \left(\tilde{\beta} \right) = \sigma^2 (X^T X)^{-1} + 0 + \sigma^2 D D^T \quad (14.62)$$

or

$$\text{Cov} \left(\tilde{\beta} \right) = \text{Cov} \left(\hat{\beta} \right) + \sigma^2 D D^T \quad (14.63)$$

and since $D D^T$ is positive semi-definite for any matrix D by the properties of the transpose, one can say that

$$\text{Cov} \left(\tilde{\beta} \right) > \text{Cov} \left(\hat{\beta} \right) \quad (14.64)$$

which implies non-minimum variance for any other unbiased estimator than the BLUE as defined above.

References

For more information, please refer to the following

- Klein, V., and Morelli, E. G., “Chapter 4 Outline of Estimation Theory,” in *Aircraft System Identification: Theory and Practice*, AIAA, 2006, pp. 75-94
- Klein, V., and Morelli, E. G., “6.1.3 Properties of Maximum Likelihood Parameter Estimates,” in *Aircraft System Identification: Theory and Practice*, AIAA, 2006, pp. 190-191
- Simon, D., “5.3 One-Step Kalman Filter Equations,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 131
- Simon, D., “8.2 Derivation of the Continuous-Time Kalman Filter,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 233-236

14.2 Introductory Least-Squares Regression

In optimal parameter estimation, one can alternatively select the parameters via the **regression problem** where one has N sets of independent data, $\vec{x}[k]$ which produced dependent samples, $\vec{y}[k]$, where $k = 1, \dots, N$. In addition, consider that one desires to optimally fit this data to a chosen **regression model**, i.e.

$$\vec{y}[k] = f_k(k, \vec{x}[k], \vec{\beta}) + \vec{\epsilon}[k] \quad (14.65)$$

where $\vec{\beta}$ contains the model parameters, also known as the **regressors**, and $\vec{\epsilon}[k]$ is the **error** random process with zero-mean, covariances $R[k]$, and uncorrelated with respect to k . The **residuals** are defined as

$$\vec{r}[k] = \vec{y}[k] - f_k(k, \vec{x}[k], \vec{\beta}) \sim \vec{\epsilon}[k] \quad (14.66)$$

To cast this in vector and matrix notation, one can redefine these in as the stacked vectors

$$\vec{y} = \begin{bmatrix} \vec{y}[1] \\ \vdots \\ \vec{y}[N] \end{bmatrix} \quad (14.67)$$

$$\vec{x} = \begin{bmatrix} \vec{x}[1] \\ \vdots \\ \vec{x}[N] \end{bmatrix} \quad (14.68)$$

$$\mathbf{f}(\vec{x}, \vec{\beta}) = \begin{bmatrix} f_1(1, \vec{x}[1], \vec{\beta}) \\ \vdots \\ f_N(N, \vec{x}[N], \vec{\beta}) \end{bmatrix} \quad (14.69)$$

$$\vec{\epsilon} = \begin{bmatrix} \vec{\epsilon}[1] \\ \vdots \\ \vec{\epsilon}[N] \end{bmatrix} \quad (14.70)$$

$$\mathbf{R} = \begin{bmatrix} R[1] & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R[N] \end{bmatrix} \quad (14.71)$$

and

$$\vec{\mathbf{r}} = \begin{bmatrix} \vec{r}[1] \\ \vdots \\ \vec{r}[N] \end{bmatrix} \quad (14.72)$$

which results in the stacked or “batch” regression model

$$\vec{\mathbf{y}} = \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}) + \vec{\epsilon} \quad (14.73)$$

Choosing the optimality criterion of least-squares of the residuals, one can form the **least-squares (LS) regression problem**, which minimizes the sum of squares of the residuals, i.e.

$$\hat{\vec{\beta}}_{LS} = \underset{\vec{\beta}}{\operatorname{argmin}} \vec{\mathbf{r}}^T \mathbf{R}^{-1} \vec{\mathbf{r}} = \underset{\vec{\beta}}{\operatorname{argmin}} \left(\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}) \right)^T \mathbf{R}^{-1} \left(\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \vec{\beta}) \right) \quad (14.74)$$

where $\hat{\vec{\beta}}_{LS}$ is the **least-squares estimator** and \mathbf{R} is a positive definite weight matrix for each sample $k = 1, \dots, N$. The term “least” in *least-squares* refers to a minimization while the term “squares” refers to a summation of multiple squared terms, in particular, the residuals. It can also be pointed out that the LS estimator minimizes the Mahalanobis distance between the samples, $\vec{\mathbf{y}}$, and the model distribution with unknown mean $\mathbf{f}(\vec{\mathbf{x}}, \vec{\beta})$ and covariance \mathbf{R} . In this case, one doesn’t explicitly require $f_{\vec{\mathbf{Y}}|\vec{\beta}}(\vec{\mathbf{y}}|\vec{\beta})$, only the mean and covariance of the regression model.

Furthermore, if the chosen regression model, $f()$, is linear, i.e.

$$\vec{\mathbf{y}}[k] = X[k] \vec{\beta} + \vec{\epsilon}[k] \quad (14.75)$$

where $X[k]$ are matrices. Then, defining the stacked matrix

$$\mathbf{X} = [X[1] \ \cdots \ X[N]] \quad (14.76)$$

one can rewrite the **linear least-squares problem** as

$$\hat{\vec{\beta}}_{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \left(\vec{\mathbf{y}} - \mathbf{X} \vec{\beta} \right)^T \mathbf{R}^{-1} \left(\vec{\mathbf{y}} - \mathbf{X} \vec{\beta} \right) \quad (14.77)$$

where $\hat{\vec{\beta}}_{LLS}$ is the **linear least-squares (LLS) parameter estimator**.

If $\mathbf{R} = \sigma^2 I$, then the LLS problem is the **ordinary least-squares (OLS) problem**. The **Gauss-Markov theorem** states that if the errors, $\vec{\epsilon}[k]$, are truly zero-mean, serially uncorrelated, and have constant covariances with respect to k , i.e., homoscedastic, then the OLS estimator is the BLUE. If also $\vec{\epsilon}[k] \sim \mathcal{N}(0, \sigma^2 I)$, then the OLS is also the MLE. If any R varies with $[k]$, then LLS problem is the **generalized least-squares (GLS) problem**. The corollary to Gauss-Markov theorem is that if $\vec{\epsilon}[k]$ are zero-mean, serially uncorrelated, and heteroscedastic, then the GLS is the BLUE. If also $\vec{\epsilon}[k] \sim \mathcal{N}(0, R[k])$, the GLS is also the MLE. Note that if \mathbf{R} are diagonal, the GLS problem is also known as the **weighted least-squares (WLS) problem**.

Ordinary Least-Squares Regression

If \mathbf{X} is square, then the OLS can be solved by simply taking the inverse of X

$$\vec{\beta}_{OLS} = \mathbf{X}^{-1} \vec{\mathbf{y}} \quad (14.78)$$

However, if \mathbf{X} is not square, then the true inverse of \mathbf{X} does not exist, then one uses the **pseudoinverse**, also known as the **Moore-Penrose inverse**, \mathbf{X}^+ , which satisfies the following properties:

- $\mathbf{X}\mathbf{X}^+\mathbf{X} = \mathbf{X}$
- $\mathbf{X}^+\mathbf{X}\mathbf{X}^+ = \mathbf{X}^+$
- $(\mathbf{X}\mathbf{X}^+)^* = \mathbf{X}\mathbf{X}^+$
- $(\mathbf{X}^+\mathbf{X})^* = \mathbf{X}^+\mathbf{X}$

It can be shown that the pseudoinverse exists for all matrices, but may not have a simple algebraic formula. However, if \mathbf{X} has linearly independent columns (i.e. full rank), then

$$\mathbf{X}^+ = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \quad (14.79)$$

which is also known as the **left pseudoinverse**.

Thus, the **ordinary least-squares (OLS) parameter estimator** can be generalized as

$$\vec{\beta}_{OLS} = \mathbf{X}^+ \vec{\mathbf{y}} \quad (14.80)$$

and if $\mathbf{X}^T \mathbf{X}$ is invertible

$$\vec{\beta}_{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{\mathbf{y}} \quad (14.81)$$

As a proof of the OLS solution, let

$$\mathcal{J}(\vec{\beta}) = (\vec{\mathbf{y}} - \mathbf{X}\vec{\beta})^T (\vec{\mathbf{y}} - \mathbf{X}\vec{\beta}) \quad (14.82)$$

Then, the minimum of $\mathcal{J}(\vec{\beta})$ occurs when its gradient is equal to $\vec{0}$.

$$\nabla \mathcal{J}(\vec{\beta}) = 2\mathbf{X}^T (\mathbf{X}\vec{\beta}_{OLS} - \vec{\mathbf{y}}) = \vec{0}^T \quad (14.83)$$

$$\mathbf{X}^T \mathbf{X}\vec{\beta}_{OLS} - \mathbf{X}^T \vec{\mathbf{y}} = \vec{0} \quad (14.84)$$

$$\mathbf{X}^T \mathbf{X}\vec{\beta}_{OLS} = \mathbf{X}^T \vec{\mathbf{y}} \quad (14.85)$$

$$\vec{\beta}_{OLS} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{\mathbf{y}} \quad (14.86)$$

which exists as long as $\mathbf{X}^T \mathbf{X}$ is left-invertible, i.e., \mathbf{X} full rank.

Solving for $\mathbf{X}^T \mathbf{X}\vec{\beta}^* = \mathbf{X}^T \vec{\mathbf{y}}$ by back substitution can suffer from numerical difficulties due to the matrix multiplication $\mathbf{X}^T \mathbf{X}$ for certain matrices. Thus, most solvers use the QR decomposition of \mathbf{X} which allows one to reform $\vec{\beta}_{OLS}$ as

$$\vec{\beta}_{OLS} = ((QR)^T QR)^{-1} (QR)^T \vec{\mathbf{y}} \quad (14.87)$$

$$\vec{\beta}_{OLS} = \left(R^T Q^T Q R \right)^{-1} R^T Q^T \vec{y} \quad (14.88)$$

$$\vec{\beta}_{OLS} = \left(R^T R \right)^{-1} R^T Q^T \vec{y} \quad (14.89)$$

$$\vec{\beta}_{OLS} = R^{-1} R^{-T} R^T Q^T \vec{y} \quad (14.90)$$

$$\vec{\beta}_{OLS} = R^{-1} Q^T \vec{y} \quad (14.91)$$

$$R \vec{\beta}_{OLS} = Q^T \vec{y} \quad (14.92)$$

Finally, it should be noted that numerical algorithms for efficiently computing the OLS solution are typically performed as follows.

1. Compute the QR decomposition on X
 - Method typically uses Householder matrices
 - Reflects vector about some hyperplane
2. Compute $\vec{c} = Q^T \vec{y}$
3. Solve $R \vec{\beta}_{OLS} = \vec{c}$ for $\vec{\beta}_{OLS}$ by back substitution

A metric that quantifies the “nearness” of $\hat{\vec{\beta}}[k]$ to $\vec{y}[k]$ is the **coefficient of determination**, R^2 . Its definition for OLS estimation follows computing the mean value of \vec{y} as

$$\vec{\mu}_{\vec{y}} = \frac{1}{N} \sum_{k=1}^N \vec{y}[k] \quad (14.93)$$

which can be stacked N times as

$$\vec{\mu}_{N,\vec{y}} = \begin{bmatrix} \vec{\mu}_{\vec{y}} \\ \vdots \\ \vec{\mu}_{\vec{y}} \end{bmatrix} \quad (14.94)$$

Then, one can analyze the **total sum of squares**, denoted by SS_{tot} , as the total squared variations in the measured output \vec{y} from $\vec{\mu}_{\vec{y}}$, denoted by SS_{tot} , defined as

$$SS_{tot} = (\vec{y} - \vec{\mu}_{N,\vec{y}})^T (\vec{y} - \vec{\mu}_{N,\vec{y}}) \quad (14.95)$$

$$SS_{tot} = \vec{y}^T \vec{y} - 2 \vec{\mu}_{N,\vec{y}}^T \vec{y} + N \vec{\mu}_{\vec{y}}^T \vec{\mu}_{\vec{y}} \quad (14.96)$$

This can be compared to the **residual sum of squares** as the sum of squared variations of the residuals between each $\vec{y}[k]$ and the estimate $\mathbf{f}(\vec{x}, \hat{\vec{\beta}})$, denoted by SS_{res} , and defined as

$$SS_{res} = (\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}))^T (\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}})) \quad (14.97)$$

For well-fit models, SS_{res} will be small relative to SS_{tot} . Specifically, the coefficient of determination, $0 \leq R^2 \leq 1$, represents the proportion of the variation in the measured output that is explained by the model, i.e.

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (14.98)$$

where values of R^2 vary from 0 to 1, where 1 represents a perfect fit to the data. Note that R^2 is usually expressed as a percentage in regression analysis.

For OLS regression, this variation can be partitioned into two parts. The first is SS_{res} which for OLS is

$$SS_{res} = (\vec{y} - \mathbf{X}\hat{\beta})^T(\vec{y} - \mathbf{X}\hat{\beta}) \quad (14.99)$$

which, by expansion of the quadratic terms, one has

$$SS_{res} = \vec{y}^T \vec{y} - 2\hat{\beta}^T \mathbf{X}^T \vec{y} + \hat{\beta}^T \mathbf{X}^T \mathbf{X} \hat{\beta} \quad (14.100)$$

and by substitution for $\hat{\beta}$,

$$SS_{res} = \vec{y}^T \vec{y} - 2\hat{\beta}^T \mathbf{X}^T \vec{y} + \hat{\beta}^T \mathbf{X}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y} \quad (14.101)$$

$$SS_{res} = \vec{y}^T \vec{y} - 2\hat{\beta}^T \mathbf{X}^T \vec{y} + \hat{\beta}^T \mathbf{X}^T \vec{y} \quad (14.102)$$

The second is the **regression sum of squares** also known as the **explained sum of squares**, denoted by SS_{reg} , and defined as the sum of squared variations of the model about the same mean value, i.e.,

$$SS_{reg} = (\mathbf{X}\hat{\beta} - \vec{\mu}_{N,\vec{y}})^T(\mathbf{X}\hat{\beta} - \vec{\mu}_{N,\vec{y}}) \quad (14.103)$$

which, by expansion of the quadratic terms, one has

$$SS_{reg} = (\mathbf{X}\hat{\beta})^T \mathbf{X}\hat{\beta} - 2\vec{\mu}_{N,\vec{y}}^T \mathbf{X}\hat{\beta} + \vec{\mu}_{N,\vec{y}}^T \vec{\mu}_{N,\vec{y}} \quad (14.104)$$

and by substitution for

$$SS_{reg} = \hat{\beta}^T \mathbf{X}^T \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y} - 2\vec{\mu}_{N,\vec{y}}^T \mathbf{X}\hat{\beta} + N\vec{\mu}_{\vec{y}}^T \vec{\mu}_{\vec{y}} \quad (14.105)$$

or

$$SS_{reg} = \hat{\beta}^T \mathbf{X}^T \vec{y} - 2\vec{\mu}_{N,\vec{y}}^T \mathbf{X}\hat{\beta} + N\vec{\mu}_{\vec{y}}^T \vec{\mu}_{\vec{y}} \quad (14.106)$$

Thus, for OLS, one has

$$SS_{reg} + SS_{res} = \vec{y}^T \vec{y} - 2\vec{\mu}_{N,\vec{y}}^T \mathbf{X}\hat{\beta} + N\vec{\mu}_{\vec{y}}^T \vec{\mu}_{\vec{y}} = SS_{tot} \quad (14.107)$$

and

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} = \frac{SS_{reg}}{SS_{tot}} = \frac{\hat{\beta}^T \mathbf{X}^T \vec{y} - 2\vec{\mu}_{N,\vec{y}}^T \mathbf{X}\hat{\beta} + N\vec{\mu}_{\vec{y}}^T \vec{\mu}_{\vec{y}}}{\vec{y}^T \vec{y} - 2\vec{\mu}_{N,\vec{y}}^T \mathbf{X}\hat{\beta} + N\vec{\mu}_{\vec{y}}^T \vec{\mu}_{\vec{y}}} \quad (14.108)$$

The covariance matrix of the OLS estimator error can be shown to be

$$\text{Cov}(\hat{\beta}) = \mathbb{E}[(\hat{\beta} - \mathbb{E}[\hat{\beta}])(\hat{\beta} - \mathbb{E}[\hat{\beta}])^T] \quad (14.109)$$

$$\text{Cov}(\hat{\beta}) = \mathbb{E}[(\hat{\beta} - \vec{\beta})(\hat{\beta} - \vec{\beta})^T] \quad (14.110)$$

$$\text{Cov}(\hat{\beta}) = \mathbb{E}[(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \vec{y} - (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\vec{y} - \vec{\epsilon})] (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\vec{y} - \vec{\epsilon})^T \quad (14.111)$$

$$\text{Cov}(\hat{\beta}) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbb{E}[\vec{\epsilon} \vec{\epsilon}^T] \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \quad (14.112)$$

$$\text{Cov}(\hat{\beta}) = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T \mathbf{R} \mathbf{X} (\mathbf{X}^T \mathbf{X})^{-1} \quad (14.113)$$

Furthermore, if the regression error is uncorrelated and has constant variance across k , i.e., $\mathbf{R} = \sigma^2 I$ then one has

$$\text{Cov}(\hat{\beta}) = \sigma^2 (\mathbf{X}^T \mathbf{X})^{-1} \quad (14.114)$$

Recall that the diagonal elements of this covariance matrix represent the parameter variances, $\sigma^2(\hat{\beta}_i)$, i.e. the standard deviations squared. Furthermore, under the assumption of Gaussian distributed parameter estimate, one can form the **100(1- α)% confidence interval** for each parameter estimate, $\hat{\beta}_i$ for $i = 1, \dots, n_\beta$, using the

$$\hat{\beta}_i + F_t^{-1}(\alpha/2, N - n_\beta) \sigma(\hat{\beta}_i) \leq \beta_i \leq \hat{\beta}_i + F_t^{-1}(1 - \alpha/2, N - n_\beta) \sigma(\hat{\beta}_i) \quad (14.115)$$

where F_t^{-1} is the inverse CDF of the t -distribution. Recall that as $N - n_\beta \rightarrow \infty$, F_t^{-1} will approach the inverse standard normal distribution, F_N^{-1} , e.g., $F_N^{-1}(0.05/2) = -1.96 \approx -2$ and $F_N^{-1}(1 - 0.05/2) = 1.96 \approx 2$.

The off-diagonal elements represent the covariance between any two parameter estimates, $\sigma^2(\hat{\beta}_i, \hat{\beta}_j)$, and can be related to the correlation coefficient between any two parameter estimates by diving by the standard deviations of each parameter, i.e.

$$\rho(\hat{\beta}_i, \hat{\beta}_j) = \frac{\sigma^2(\hat{\beta}_i, \hat{\beta}_j)}{\sigma(\hat{\beta}_i)\sigma(\hat{\beta}_j)} \quad (14.116)$$

In the practice of least-squares regression, often one does not know if the samples are uncorrelated with respect to the time step, k . One method for checking this correlation is by computing the **residual auto-correlation function** of the observed residuals between the fitted model and the samples, i.e.

$$R_{\vec{r}\vec{r}}[j] = \frac{1}{N} \sum_{k=1}^{N-j} \vec{r}[j] \vec{r}^T[k+j] \quad (14.117)$$

for **frequency indexes** $j = 0, \dots, N - 1$, i.e. the different possible time separations for the residuals. As this regression approach to parameter estimation assumes the sampling process is WSS, if the residuals are completely uncorrelated, then it should be that $R_{\vec{r}\vec{r}}[j]$ is approximately a zero matrix for $j \neq 0$. Although the estimated residuals will not be exactly zero, but one should expect $100(1 - \alpha)\%$ of the i^{th} residual element would be randomly distributed within $\pm F_t^{-1}(1 - \alpha/2, N - n_y) \sqrt{R_{r_i r_i}[0]} N$ by a Gaussian assumption on the residual auto-correlation function distribution.

Generalized Least-Squares Regression

The GLS estimate can be considered as the OLS on scaled and “de-correlated” observations. To see this consider the Cholesky decomposition of \mathbf{R} , i.e.

$$\mathbf{R} = CC^T \quad (14.118)$$

Next, multiply the observation model by C^{-1} on both sides, i.e.

$$C^{-1}\vec{y} = C^{-1}\mathbf{X}\vec{\beta} + C^{-1}\vec{\epsilon} \quad (14.119)$$

Then, one can form a “new” model

$$\vec{y}^* = \mathbf{X}^*\vec{\beta} + \vec{\epsilon}^* \quad (14.120)$$

where

$$\vec{y}^* = C^{-1}\vec{y} \quad (14.121)$$

$$\mathbf{X}^* = C^{-1}\mathbf{X} \quad (14.122)$$

$$\vec{\epsilon}^* = C^{-1}\vec{\epsilon} \quad (14.123)$$

For this model, one can show that

$$\text{Cov}(\vec{\epsilon}^*) = \text{Cov}(C^{-1}\vec{\epsilon}) \quad (14.124)$$

$$\text{Cov}(\vec{\epsilon}^*) = C^{-1}\text{Cov}(\vec{\epsilon})C^{-T} \quad (14.125)$$

or

$$\text{Cov}(\vec{\epsilon}^*) = C^{-1}RC^{-T} \quad (14.126)$$

and substituting for R from the Cholesky decomposition, one has

$$\text{Cov}(\vec{\epsilon}^*) = C^{-1}CC^TC^{-T} \quad (14.127)$$

$$\text{Cov}(\vec{\epsilon}^*) = I \quad (14.128)$$

Thus, substituting back for the “new” parameters into the expression for $\hat{\vec{\beta}}_{OLS}$, one has

$$\hat{\vec{\beta}}_{GLS} = (\mathbf{X}^{*T}\mathbf{X}^*)^{-1}\mathbf{X}^{*T}\vec{y}^* \quad (14.129)$$

$$\hat{\vec{\beta}}_{GLS} = (\mathbf{X}^TC^{-T}C^{-1}\mathbf{X})^{-1}\mathbf{X}^TC^{-T}C^{-1}\vec{y} \quad (14.130)$$

Thus, by substitution, the **generalized least-squares (GLS) parameter estimator** is

$$\hat{\vec{\beta}}_{GLS} = (\mathbf{X}^T\mathbf{R}^{-1}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}^{-1}\vec{y} \quad (14.131)$$

where the covariance is given by

$$\text{Cov}(\hat{\vec{\beta}}_{GLS}) = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{R}^{-1}\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1} \quad (14.132)$$

References

For more information, please refer to the following

- Klein, V., and Morelli, E. G., “Chapter 5.1 Ordinary Least Squares,” in *Aircraft System Identification: Theory and Practice*, AIAA, 2006, pp. 97-131
- Klein, V., and Morelli, E. G., “Chapter 5.2 Generalized Least Squares,” in *Aircraft System Identification: Theory and Practice*, AIAA, 2006, pp.132-136
- Sarkka, S., “3.1 Batch linear regression,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 27-30
- Simon, D., “3 Least squares estimation,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 131

14.3 Advanced Batch Least-Squares Methods

Nonlinear Least-Squares Regression

For nonlinear $f()$ and with an optimality criterion of least-squares, one can form the **nonlinear least-squares (NLS) regression problem**, $\hat{\beta}_{NLS}$, which minimizes the sum of squares of the residuals, i.e.

$$\hat{\beta}_{NLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \vec{r}^T \mathbf{R}^{-1} \vec{r} = \underset{\vec{\beta}}{\operatorname{argmin}} \left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right)^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right) \quad (14.133)$$

where $\hat{\beta}_{NLS}$ is the **nonlinear least-squares (NLS) estimator**. Any local minimum for this optimization, if one exists, is found by setting the derivative of the squared residuals to zero, i.e.

$$2 \vec{r}^T \frac{\partial \vec{r}}{\partial \vec{\beta}} = 0 \quad (14.134)$$

or, in terms of the regression model Jacobian, $\frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}}$, one has

$$-2 \vec{r}^T \frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} = 0 \quad (14.135)$$

where the optimal estimate, $\hat{\beta}_{NLS}$, will be the one (or multiple) values of $\vec{\beta}$ which accomplish this.

However, one cannot often find an analytical solution to the NLS estimation problem, thus, one must use numerical methods to find the optimal value. Such methods begin with some initial estimate, $\hat{\beta}_0$, and then use an iterative procedure to refine the parameter estimate, i.e.

$$\hat{\beta}_{i+1} = \hat{\beta}_i + \vec{\Delta}_i \quad (14.136)$$

where i is the iteration number and $\vec{\Delta}_i$ is the **search vector** at iteration i . Then, after some convergence criteria for the search vector, one can approximate the NLS estimator by

$$\hat{\beta}_{NLS} \approx \hat{\beta}_i \quad \text{for } \frac{\|\vec{\Delta}_i\|}{\|\hat{\beta}_i\|} < \delta \quad (14.137)$$

where a typical value for δ would be 0.001 which would require a precision of 0.1%. When used in this form, the NLS estimator is a type of **iterative least-squares (ILS) estimator**.

One of the simplest methods for computing $\vec{\Delta}_i$ is the **Gauss-Newton algorithm (GNA)** which approximates the observation model by a first-order Taylor series expansion, i.e.

$$\mathbf{f}(\vec{x}, \vec{\beta}) \approx \mathbf{f}(\vec{x}, \hat{\beta}_i) + \frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} \Big|_{\vec{\beta}=\hat{\beta}_i} \vec{\Delta}_i \vec{\beta} \quad (14.138)$$

where

$$\vec{\Delta}_i = \vec{\beta} - \hat{\beta}_i \quad (14.139)$$

Next, defining the Jacobian of the regression model as

$$\mathbf{J} = \frac{\partial \mathbf{f}(\vec{x}, \vec{\beta})}{\partial \vec{\beta}} \Big|_{\vec{\beta}=\hat{\vec{\beta}}_i} \quad (14.140)$$

one can rewrite $\vec{r} = \vec{y} - \mathbf{f}(\vec{x}, \vec{\beta})$ as

$$\vec{r} = \vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) + \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) - \mathbf{f}(\vec{x}, \vec{\beta}) \quad (14.141)$$

or

$$\vec{r} \approx \vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) - \mathbf{J}\vec{\Delta}_i \quad (14.142)$$

Then, setting the gradient of this approximation for the squared residuals to zero, one has

$$-2\vec{r}^T \frac{\partial f}{\partial \vec{\beta}} = -2 \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) - \mathbf{J}\vec{\Delta}_i \right)^T \mathbf{J} = 0 \quad (14.143)$$

which can be rewritten as

$$\mathbf{J}^T \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) - \mathbf{J}\vec{\Delta}_i \right) = 0 \quad (14.144)$$

and by rearranging, one obtains the OLS solution for the search vector

$$\vec{\Delta}_i = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) \right) \quad (14.145)$$

Furthermore, if the residuals have an expected covariance matrix, \mathbf{R} , then one can alternatively use the GLS solution as

$$\vec{\Delta}_i = (\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) \right) \quad (14.146)$$

An important part of this numerical method is the initial parameter estimates and the problem of divergence. Since divergence in the GNA often occurs, one often uses the **Levenburg-Marquardt algorithm (LMA)**, also known as **damped least-squares (DLS)**, as a trust-region augmentation to the GNA. In this case, one includes a damping factor, ζ_i , also known as **Marquardt parameter** in the search vector calculation as

$$\vec{\Delta}_i = (\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J} + \zeta_i I)^{-1} \mathbf{J}^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) \right) \quad (14.147)$$

where this second term combines the GNA with the direction of **steepest gradient**, i.e.

$$\vec{\Delta}_i = \frac{1}{\zeta_i} \mathbf{J}^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) \right) \quad (14.148)$$

which the LMA approximates if $\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J} \ll \zeta_i I$. The LMA can be further improved through Fletcher's **modified LMA**

$$\vec{\Delta}_i = (\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J} + \zeta_i \text{diag}(\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J}))^{-1} \mathbf{J}^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_i) \right) \quad (14.149)$$

where $\text{diag}(\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J})$ selects the diagonal elements of $\mathbf{J}^T \mathbf{R}^{-1} \mathbf{J}$. This modification slows down convergence in the direction of small gradients by adjusting more where the gradient of $f()$ is smaller.

In either case of LMA, ζ_i is heuristically adjusted at each iteration which is intrinsic to the trust-region method. An effective heuristic known as **delayed gratification** consists of increasing ζ_i by a small amount for iterations where $\vec{r}^T \vec{r}$ increases, and decreasing ζ_i by a large amount for iterations where $\vec{r}^T \vec{r}$ decreases. This heuristic slows down convergence which avoids converging too quickly in the beginning of optimization. An increase by a factor of 2 and a decrease by a factor of 3 are effective for most problems. Other trust-region methods can also be used which solve a sub-problem for the optimal damping factor which are discussed in the next section.

Constrained Least-Squares Regression

Trust-region methods, also known as **restricted-step methods**, can be used for both unconstrained and constrained least-squares regression problems. The **constrained least-squares (CLS) regression problem** can be stated as

$$\begin{aligned}\hat{\vec{\beta}}^{CLS} &= \underset{\vec{\beta} \in X}{\operatorname{argmin}} \quad \left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right)^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right) \\ \text{subject to: } l_{\beta} &\leq \vec{\beta} \leq u_{\beta}\end{aligned}\tag{14.150}$$

where l_{β} and u_{β} are the lower and upper bounds on the parameter estimate, respectively. This section will discuss how trust-region methods solve the CLS problem at a high level.

To understand trust-region methods, consider the general minimization search problem where one desires to improve, i.e. move from $\hat{\vec{\beta}}_i$ to some $\hat{\vec{\beta}}_{i+1}$ which reduces the least-squares function. Trust-region methods approximate the least-squares function, $\left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right)^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right)$ with a simpler function, $q(\vec{y}, \vec{x}, \vec{\beta})$ in some neighborhood about $\hat{\vec{\beta}}_i$, then solving for the value where the least-squares function approximation does decrease, i.e. the optimal **trial step vector**, $\vec{\Delta}_i$, within that neighborhood, i.e. **trust-region**. However, one must also check that the obtained trial step vector can be “trusted,” i.e. that

$$\left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right)^T \mathbf{R}^{-1} \left(\vec{y} - \mathbf{f}(\vec{x}, \vec{\beta}) \right) < q(\vec{y}, \vec{x}, \hat{\vec{\beta}}_i + \vec{\Delta}_i)\tag{14.151}$$

where if this fails to hold, then the trust-region must be adjusted and a new trial step vector must be found. Thus, different trust-region methods use different approximations, $q()$, different trial step vector solvers in the trust-region, and different trust-region adjustments. Standard trust-region methods use a quadratic form for the least-squares approximation, e.g. the LMA, which defines an ellipsoidal trust-region, quadratic programming solvers, and standard trust-region adjustments similar to the delayed gratification heuristic above.

Quadratic Programming

As an aside, the **quadratic programming (QP) problem** can be stated as

$$\begin{aligned}\vec{\beta}^{QP} &= \underset{\vec{\beta}}{\operatorname{argmin}} \quad \frac{1}{2} \vec{\beta}^T Q \vec{\beta} + c^T \vec{\beta} \\ \text{subject to: } A \vec{\beta} &\leq \vec{b}\end{aligned}\tag{14.152}$$

where the constraint is a vector-defined component-wise inequality. QP problems appear in different problems. One is minimizing a function $f()$ in a local neighborhood about a point $\vec{\beta}_0$, e.g. a trust-region, one can set Q to the Hessian matrix at that point, $H(\vec{\beta}_0)$, and \vec{c} to its gradient, $\nabla f(\vec{\beta}_0)$.

As a special case, when $Q = Q^T > 0$, a QP problem is equivalent to the LLS problem. To see this, consider the LLS minimization written as

$$\vec{\beta}_{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \quad (\vec{y} - \mathbf{X}\vec{\beta})^T (\vec{y} - \mathbf{X}\vec{\beta}) \quad (14.153)$$

$$\vec{\beta}_{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \quad \vec{y}^T \vec{y} - \vec{y}^T \mathbf{X} \vec{\beta} - \vec{\beta}^T \mathbf{X}^T \vec{y} + \vec{\beta}^T \mathbf{X}^T \mathbf{X} \vec{\beta} \quad (14.154)$$

which is equivalent to minimizing only over terms with $\vec{\beta}$, multiplying the expression by $\frac{1}{2}$, and that the order of the middle terms produces the same result, i.e.

$$\vec{\beta}_{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \quad \frac{1}{2} \vec{\beta}^T \mathbf{X}^T \mathbf{X} \vec{\beta} - \vec{y}^T \mathbf{X} \vec{\beta} \quad (14.155)$$

which by the Cholesky decomposition $Q = \mathbf{X}^T \mathbf{X}$ and defining $\vec{c} = -\mathbf{X}^T \vec{y}$, one has

$$\vec{\beta}_{LLS} = \underset{\vec{\beta}}{\operatorname{argmin}} \quad \frac{1}{2} \vec{\beta}^T Q \vec{\beta} + \vec{c}^T \vec{\beta} \quad (14.156)$$

QP programming solvers can be derived explicitly for equality constraints and for $Q > 0$ similar to the least-squares solver, i.e. the ellipsoid method solves the problem in (weakly) polynomial time. However if Q is indefinite or has even one negative eigenvalue, then the problem is NP-hard.

Discrete-Time LQR as Linear Least-Squares Problem

The unconstrained finite-horizon discrete-time LQR can be formulated as a LLS problem. Consider the unconstrained finite-horizon discrete-time LQ OCP written as

$$\begin{aligned} \vec{u}_{\text{opt}}[k] &= \underset{\vec{u}[k] \text{ for } k=0, \dots, N-1}{\operatorname{argmin}} \quad \mathcal{J} = \vec{x}^T[N] E \vec{x}[N] + \sum_{k=0}^{N-1} \vec{x}^T[k] Q \vec{x}[k] + \vec{u}^T[k] R \vec{u}[k] \\ &\text{subject to: } \vec{x}[k+1] = F \vec{x}[k] + G \vec{u}[k] \\ &\text{initial condition: } \vec{x}[0] = \vec{x}_0 \end{aligned} \quad (14.157)$$

where S has been removed from the cost function for simplification of the following derivations, although it can be included in the LLS solution for the discrete-time LQR as well.

Based on the discrete-time linear dynamics and initial condition, one can write out the general solution of the state as a “large” linear function, i.e.

$$\begin{bmatrix} \vec{x}[0] \\ \vdots \\ \vec{x}[N] \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & \cdots & 0 \\ G & 0 & 0 & \cdots & 0 \\ FG & G & 0 & \cdots & 0 \\ F^2G & FG & G & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ F^{N-1}G & F^{N-2}G & F^{N-3}G & \cdots & G \end{bmatrix} \begin{bmatrix} \vec{u}[0] \\ \vdots \\ \vec{u}[N-1] \end{bmatrix} + \begin{bmatrix} I \\ F \\ \vdots \\ F^N \end{bmatrix} \vec{x}_0 \quad (14.158)$$

which one can rewrite succinctly by defining the above vectors and matrices as

$$\vec{x} = \mathbf{G}\vec{u} + \mathbf{F}\vec{x}_0 \quad (14.159)$$

Then, using these new terms and the following block diagonal matrices

$$\mathbf{Q} = \begin{bmatrix} Q & \cdots & 0 & 0 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & Q & 0 \\ 0 & \cdots & 0 & E \end{bmatrix} \quad (14.160)$$

and

$$\mathbf{R} = \begin{bmatrix} R & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R \end{bmatrix} \quad (14.161)$$

the cost function for the discrete-time LQ OCP can be rewritten using vector notation as

$$\mathcal{J} = \vec{x}^T \mathbf{Q} \vec{x} + \vec{u}^T \mathbf{R} \vec{u} \quad (14.162)$$

and substituting for \vec{x} from before, one can write

$$\mathcal{J} = (\mathbf{G}\vec{u} + \mathbf{F}\vec{x}_0)^T \mathbf{Q} (\mathbf{G}\vec{u} + \mathbf{F}\vec{x}_0) + \vec{u}^T \mathbf{R} \vec{u} \quad (14.163)$$

By using the definition of the square root of positive semi-definite matrices, i.e., $Q = Q^{\frac{1}{2}}Q^{\frac{1}{2}}$, the cost function can be rewritten as

$$\mathcal{J} = (\mathbf{G}\vec{u} + \mathbf{F}\vec{x}_0)^T \mathbf{Q}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}} (\mathbf{G}\vec{u} + \mathbf{F}\vec{x}_0) + \vec{u}^T \mathbf{R}^{\frac{1}{2}} \mathbf{R}^{\frac{1}{2}} \vec{u} \quad (14.164)$$

and using the definition of vector norms, one has

$$\mathcal{J} = \left\| \mathbf{Q}^{\frac{1}{2}} \mathbf{G} \vec{u} + \mathbf{Q}^{\frac{1}{2}} \mathbf{F} \vec{x}_0 \right\|_2^2 + \left\| \mathbf{R}^{\frac{1}{2}} \vec{u} \right\|_2^2 \quad (14.165)$$

which is a LLS problem. This form also uses the fact that \mathbf{Q} and \mathbf{R} are symmetric since E , Q , and R are as well.

Finally, taking the gradient for L^2 -norms squared with respect to \vec{u} ,

$$\nabla \mathcal{J} = 2 \left(\mathbf{Q}^{\frac{1}{2}} \mathbf{G} \right)^T \left(\mathbf{Q}^{\frac{1}{2}} \mathbf{G} \vec{u} + \mathbf{Q}^{\frac{1}{2}} \mathbf{F} \vec{x}_0 \right) + 2 \mathbf{R} \vec{u} \quad (14.166)$$

and solving for $\nabla \mathcal{J} = 0$,

$$2 \mathbf{G}^T \mathbf{Q}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}} \mathbf{G} \vec{u}^{\text{opt}} + 2 \mathbf{G}^T \mathbf{Q}^{\frac{1}{2}} \mathbf{Q}^{\frac{1}{2}} \mathbf{F} \vec{x}_0 + 2 \mathbf{R} \vec{u}^{\text{opt}} = 0 \quad (14.167)$$

$$\mathbf{G}^T \mathbf{Q} \mathbf{G} \vec{u}^{\text{opt}} + \mathbf{G}^T \mathbf{Q} \mathbf{F} \vec{x}_0 + \mathbf{R} \vec{u}^{\text{opt}} = 0 \quad (14.168)$$

$$(\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R}) \vec{u}^{\text{opt}} = -\mathbf{G}^T \mathbf{Q} \mathbf{F} \vec{x}_0 \quad (14.169)$$

$$\vec{u}_{\text{opt}} = - \left(\mathbf{G}^T \mathbf{Q} \mathbf{G} + \mathbf{R} \right)^{-1} \mathbf{G}^T \mathbf{Q} \mathbf{F} \vec{x}_0 \quad (14.170)$$

which provides a method for solving the entire problem at once instead of recursively through the dynamic Riccati equation.

First, it should be noted that the recursive solution may be faster computationally due to the large matrix inversions that are necessary in the LLS approach; however the **sparsity**, i.e. the number of 0 entries, in the **Q** and **R** matrices can make this inversion faster. Second, it should also be noted that the LLS solution can be adjusted for linear time-varying (LTV) systems, i.e. F , G , E , Q , and R matrices vary with k , since this will only change the values within **F**, **G**, **Q** and **R**. Third, the relationship of LLS to QP allows for straightforward computation of the constrained discrete-time LQR through QP methods.

References

For more information, please refer to the following

- Klein, V., and Morelli, E. G., “Chapter 5.3 Nonlinear Least Squares,” in *Aircraft System Identification: Theory and Practice*, AIAA, 2006, pp. 137

14.4 Introductory Bayesian Optimal Parameter Estimation

Another approach to the optimal parameter estimation problem is from a Bayesian inference perspective. Here one considers that \vec{y} are samples of a random vector \vec{Y} which depend on some *random* parameter vector, $\vec{\beta}$, which has taken some realization $\vec{\beta}$. In this case, one must consider the joint probabilities of \vec{Y} and $\vec{\beta}$ which can be modeled by some joint PDF, $f_{\vec{Y}, \vec{\beta}}(\vec{y}, \vec{\beta})$. However, as one observes the realizations of \vec{Y} , one is particularly concerned with the **posterior PDF**, also known as the *a posteriori* PDF, i.e., $f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y})$, which is related to the joint PDF by

$$f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y}) = \frac{f_{\vec{Y}, \vec{\beta}}(\vec{y}, \vec{\beta})}{f_{\vec{Y}}(\vec{y})} \quad (14.171)$$

which simply restates the definition of conditional probabilities. Furthermore, one can rewrite this equation as

$$f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y}) = \frac{f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta}) f_{\vec{\beta}}(\vec{\beta})}{f_{\vec{Y}}(\vec{y})} \quad (14.172)$$

which, by the law of total probability, i.e.

$$f_{\vec{Y}}(\vec{y}) = \int f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta}) f_{\vec{\beta}}(\vec{\beta}) d\vec{\beta} \quad (14.173)$$

one has by Bayes' rule

$$f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y}) = \frac{f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta}) f_{\vec{\beta}}(\vec{\beta})}{\int f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta}) f_{\vec{\beta}}(\vec{\beta}) d\vec{\beta}} \quad (14.174)$$

Thus, Bayesian parameter estimation can be seen as using three sources of information, the **prior PDF**, also known as the *a priori* PDF, of the parameter vector, $f_{\vec{\beta}}(\vec{\beta})$, the likelihood function, $f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta})$ or $\mathcal{L}(\vec{\beta}|\vec{y})$, and the **model evidence**, $f_{\vec{Y}}(\vec{y})$, also known as the **marginal likelihood**.

Bayes Optimal Parameter Estimators

A formal way to define Bayes optimality is the so-called **Bayes estimator (BE)** where one minimizes the **Bayes risk** defined as the posterior expected value of a selected loss function, \mathcal{J} , i.e., for the **Bayes parameter estimator**

$$\hat{\vec{\beta}}_{BE} = \operatorname{argmin}_{\vec{\beta}} \mathbb{E}_{\vec{B}, \vec{Y}} [\mathcal{J}(\vec{\beta}, \hat{\vec{\beta}})] \quad (14.175)$$

where the expectation is taken over the joint distribution of \vec{B} and \vec{Y} . A common optimal Bayes parameter estimator is the **maximum *a posteriori* (MAP) estimator**, i.e., the mode of the posterior,

$$\hat{\vec{\beta}}_{MAP} = \operatorname{argmax}_{\vec{\beta}} f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) \quad (14.176)$$

and can be shown to select the negative Dirac delta error for the loss function, i.e.,

$$\hat{\vec{\beta}}_{MAP} = \operatorname{argmin}_{\hat{\vec{\beta}}} \mathbb{E}_{\vec{B}, \vec{Y}} [-\delta(\vec{\beta} - \hat{\vec{\beta}})] \quad (14.177)$$

which by Bayes' rule can be rewritten as

$$\hat{\vec{\beta}}_{MAP} = \operatorname{argmax}_{\vec{\beta}} \frac{f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta}) f_{\vec{B}}(\vec{\beta})}{f_{\vec{Y}}(\vec{y})} \quad (14.178)$$

which is equivalent to

$$\hat{\vec{\beta}}_{MAP} = \operatorname{argmax}_{\vec{\beta}} f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta}) f_{\vec{B}}(\vec{\beta}) \quad (14.179)$$

v It should be noted if the prior PDF is uniform across all values of $\vec{\beta}$, i.e., the prior is **non-informative** of $\vec{\beta}$, then, the MAP estimator is equivalent to the MLE, i.e.,

$$\hat{\vec{\beta}}_{MLE} = \operatorname{argmax}_{\vec{\beta}} f_{\vec{Y}|\vec{B}}(\vec{y}|\vec{\beta}) \quad (14.180)$$

Another common Bayes estimator is the **minimum mean-square error (MMSE) estimator** which selects the squared error for the loss function, i.e.,

$$\hat{\vec{\beta}}_{MMSE} = \operatorname{argmin}_{\hat{\vec{\beta}}} \mathbb{E}_{\vec{B}, \vec{Y}} [(\hat{\vec{\beta}} - \vec{\beta})^T (\hat{\vec{\beta}} - \vec{\beta})] \quad (14.181)$$

and, for finite means and variances, is minimized by the mean of the posterior, i.e.,

$$\hat{\vec{\beta}}_{MMSE} = \mathbb{E}_{\vec{B}|\vec{Y}}[\vec{\beta}] = \int \cdots \int \vec{\beta} f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) d\vec{\beta} \quad (14.182)$$

Thus, the $\hat{\vec{\beta}}_{MMSE} = \hat{\vec{\beta}}_{MAP}$, if $f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y})$ is elliptically symmetric about a single peak.

Lastly, the **median (Mdn) estimator** is

$$\hat{\vec{\beta}}_{Mdn} = \inf \left\{ \hat{\vec{\beta}} : \int_{-\infty}^{\hat{\vec{\beta}}} f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) d\vec{\beta} \geq 0.5 \right\} \quad (14.183)$$

and can be shown to select the absolute value error for the loss function, i.e.,

$$\hat{\vec{\beta}}_{Mdn} = \operatorname{argmin}_{\hat{\vec{\beta}}} \mathbb{E}_{\vec{B}, \vec{Y}} [| \hat{\vec{\beta}} - \vec{\beta} |] \quad (14.184)$$

Conjugate Prior Parameter Estimation

Given a particular likelihood model, one can obtain a closed-form expression for the posterior distribution if the posterior and the prior distributions belong in the same distribution family. In this case, the **conjugate prior** is defined for the specific likelihood function and the known distribution parameters of the conjugate prior and corresponding posterior are the **hyper-parameters** as opposed to the estimated parameter $\vec{\beta}$. In this context, one may also have N samples of \vec{Y} , i.e., $\vec{y}_1, \dots, \vec{y}_N$. Four common likelihood functions with their conjugate priors and corresponding posteriors for aerospace vehicle perception systems are given here without proof.

For the likelihood function given by

$$f_{Y_i|B}(y_i|\beta) = f_{Pois}(y_i;\beta) \quad (14.185)$$

with unknown rate β , a conjugate prior PDF for the rate is

$$f_B(\beta) = f_\Gamma(\beta; \alpha, \theta) \quad (14.186)$$

with the corresponding posterior PDF for the rate as

$$f_{B|Y_1, \dots, Y_N}(\beta|y_1, \dots, y_N) = f_\Gamma(\beta; \alpha + \sum_{i=1}^N y_i, \theta + N) \quad (14.187)$$

For the likelihood function given by

$$f_{\vec{Y}_i|\vec{\beta}}(\vec{y}_i|\vec{\beta}) = f_N(\vec{y}; \vec{\beta}, \Sigma_y) \quad (14.188)$$

and denoting the **sample mean** as

$$\vec{\mu}_y = N^{-1} \sum_{i=1}^N \vec{y}_i \quad (14.189)$$

one has

$$f_{\vec{M}_y|\vec{\beta}}(\vec{\mu}_y|\vec{\beta}) = f_N(\vec{\mu}_y; \vec{\beta}, N^{-1}\Sigma_y) \quad (14.190)$$

with unknown mean, $\vec{\beta}$, and known covariance, Σ_y , a conjugate prior PDF for the mean is

$$f_{\vec{B}}(\vec{\beta}) = f_N(\beta; \vec{\mu}_0, \Sigma_0) \quad (14.191)$$

with the corresponding posterior PDF as

$$f_{\vec{B}|\vec{Y}_1, \dots, \vec{Y}_N}(\vec{\beta}|\vec{y}_1, \dots, \vec{y}_N) = f_N(\vec{\beta}; \vec{\mu}, \Sigma) \quad (14.192)$$

with

$$\Sigma = (\Sigma_0^{-1} + N\Sigma_y^{-1})^{-1} \quad (14.193)$$

and

$$\vec{\mu} = \Sigma \left(\Sigma_0^{-1} \vec{\mu}_0 + \Sigma_y^{-1} \vec{\mu}_y \right) \quad (14.194)$$

For the likelihood function given by

$$f_{\vec{Y}_i|B}(\vec{y}_i|\beta) = f_N(\vec{y}_i; \vec{\mu}, \beta) \quad (14.195)$$

and denoting the **sample covariance** as

$$\Sigma_y = \frac{1}{N-1} \sum_{i=1}^N (\vec{y}_i - \vec{\mu}_y)(\vec{y}_i - \vec{\mu}_y)^T \quad (14.196)$$

with known mean, $\vec{\mu}$ and unknown covariance, β , a conjugate prior PDF for the covariance is

$$f_B(\beta) = f_{IW}(\beta; \Phi_0, v_0) \quad (14.197)$$

with the corresponding posterior PDF as

$$f_{B|\vec{Y}_1, \dots, \vec{Y}_N}(B|\vec{y}_1, \dots, \vec{y}_N) = f_{IW}(\beta; \Phi, v_0 + N) \quad (14.198)$$

with

$$\Phi = \Phi_0 + (N-1)\Sigma_y \quad (14.199)$$

For the likelihood function given by

$$f_{\vec{Y}_i|\vec{B}_\mu, B_\Sigma}(\vec{y}_i|\vec{\beta}_\mu, \beta_\Sigma) = f_N(\vec{y}_i; \vec{\beta}_\mu, \beta_\Sigma) \quad (14.200)$$

and if $\vec{\mu}_y$ is the sample mean and Σ_y as the sample scatter matrix, with unknown mean, $\vec{\beta}_\mu \in \mathbb{R}^{n_\beta}$ and unknown covariance, $\beta_\Sigma \in \mathbb{R}^{n_\beta \times n_\beta}$, a conjugate prior PDF is

$$f_{\vec{B}_\mu, B_\Sigma}(\vec{\beta}_\mu, \beta_\Sigma) = f_{NIW}(\vec{\beta}_\mu, \beta_\Sigma; \vec{\mu}_0, \lambda_0, \Phi_0, v_0) \quad (14.201)$$

with the corresponding posterior PDF as

$$f_{\vec{B}_\mu, B_\Sigma|\vec{Y}_1, \dots, \vec{Y}_N}(\vec{\beta}_\mu, \beta_\Sigma|\vec{y}_1, \dots, \vec{y}_N) = f_{NIW}(\vec{\beta}_\mu, \beta_\Sigma; \vec{\mu}, \lambda_0 + N, \Phi, v_0 + N) \quad (14.202)$$

with

$$\vec{\mu} = \frac{N}{\lambda_0 + N} \vec{\mu}_y + \frac{\lambda_0}{\lambda_0 + N} \vec{\mu}_0 \quad (14.203)$$

and

$$\Phi = \Phi_0 + (N-1)\Sigma_y + \frac{\lambda_0 N}{\lambda_0 + N} (\vec{\mu}_y - \vec{\mu}_0)(\vec{\mu}_y - \vec{\mu}_0)^T \quad (14.204)$$

Related to this case is the univariate likelihood function given by

$$f_{Y_i|B}(y_i|\beta) = f_N(y_i; \beta_\mu, \beta_\sigma^2) \quad (14.205)$$

and denoting the sample variance as

$$\sigma_y^2 = \frac{1}{N-1} \sum_{i=1}^N (y_i - \mu_y)^2 \quad (14.206)$$

with unknown mean, β_μ and unknown variance, β_σ^2 , a conjugate prior PDF is

$$f_{B_\mu, B_\sigma^2}(\beta_\mu, \beta_\sigma^2) = f_{N\chi^2}(\beta_\mu, \beta_\sigma^2; \mu_0, \lambda_0, \sigma_0^2, \nu_0) \quad (14.207)$$

with the corresponding posterior PDF as

$$f_{B_\mu, B_\sigma^2 | \vec{Y}_1, \dots, \vec{Y}_N}(\beta_\mu, \beta_\sigma^2 | \vec{y}_1, \dots, \vec{y}_N) = f_{N\chi^2}(\beta_\mu, \beta_\sigma^2; \mu, \lambda_0 + N, \sigma^2, \nu_0 + N) \quad (14.208)$$

with

$$\mu = \frac{N}{\lambda_0 + N} \vec{\mu}_y + \frac{\lambda_0}{\lambda_0 + N} \vec{\mu}_0 \quad (14.209)$$

and

$$\sigma^2 = \frac{\nu_0}{\nu_0 + N} \sigma_0^2 + \frac{N - 1}{\nu_0 + N} \sigma_y^2 + \frac{\lambda_0 N}{(\lambda_0 + N)(\nu_0 + N)} (\mu_y - \mu_0)^2 \quad (14.210)$$

Bayesian Least-Squares Regression

The **Bayesian regression problem** considers a regression model as

$$\vec{y} = f(\vec{x}, \vec{\beta}) + \vec{\epsilon} \quad (14.211)$$

with the $\mathbb{E}[\vec{\epsilon}] = 0$ and covariance $\mathbb{E}[\vec{\epsilon}\vec{\epsilon}^T] = R$, but also has some *a priori* information about the possible values of $\vec{\beta}$ represented by the prior PDF defined as $\vec{\beta}_0 \sim f_{\vec{B}}(\vec{\beta})$ and can be regarded as user-defined prior knowledge or due to previous parameter estimates. Due to this statistical nature, the objective in Bayesian least-squares regression is to minimize the sum of the estimator errors, i.e.

$$\hat{\vec{\beta}}_{BLS} = \underset{\hat{\vec{\beta}}}{\operatorname{argmin}} \mathbb{E}_{\vec{B}, \vec{Y}} [(\vec{\beta} - \hat{\vec{\beta}})^T (\vec{\beta} - \hat{\vec{\beta}})] \quad (14.212)$$

where $\hat{\vec{\beta}}_{BLS}$ is the **Bayesian least-squares (BLS) parameter estimator** and is equivalent to the MMSE estimator.

Thus, for finite means and variances, one has

$$\hat{\vec{\beta}}_{BLS} = \mathbb{E}_{\vec{B} | \vec{Y}} [\vec{\beta}] \quad (14.213)$$

BLS is also known as **recursive least-squares** as this construction allows one to recursively process sequential samples, $\vec{y}[k]$ with $k = 1, 2, \dots$, and recursively update the estimate $\hat{\vec{\beta}}[k]$. This is opposed to previous least-squares methods which require “batch” processing of the data samples using stacked vectors and matrices. The **Bayesian linear regression problem** occurs when one has a linear-Gaussian regression model

$$\vec{y} = X\vec{\beta} + \vec{\epsilon} \quad (14.214)$$

with zero-mean multivariate Gaussian errors, $\vec{\epsilon} \sim \mathcal{N}(0, R)$. This analysis typically has two different cases, R is either known or unknown. Both cases typically utilize conjugate priors in their Bayes recursions.

Bayesian Linear-Gaussian Regression

When R is known, one can specify a multivariate Gaussian conjugate prior PDF, $\vec{\beta} \sim \mathcal{N}(\vec{\beta}; \vec{\mu}_0, \Sigma_0)$ and consider a transformed sample as $\vec{z} = X^+ \vec{y}$ which produces a transformed regression model as

$$\vec{z} = \vec{\beta} + X^+ \vec{\epsilon} \quad (14.215)$$

and a transformed likelihood function given by

$$f_{\vec{Z}|\vec{\beta}}(\vec{z}|\vec{\beta}) = f_N(\vec{z}; \vec{\beta}, X^+ R X^{+T}) \quad (14.216)$$

with conjugate prior

$$f_{\vec{\beta}}(\vec{\beta}) = f_N(\vec{\beta}; \vec{\mu}_0, \Sigma_0) \quad (14.217)$$

with corresponding transformed posterior PDF as

$$f_{\vec{\beta}|\vec{Z}}(\vec{\beta}|\vec{z}) = f_N(\vec{\beta}; \hat{\vec{\beta}}, \Sigma_{\vec{\beta}}) \quad (14.218)$$

with mean

$$\hat{\vec{\beta}} = \Sigma_{\vec{\beta}} \left(X^T R^{-1} X \vec{z} + \Sigma_0^{-1} \vec{\mu}_0 \right) \quad (14.219)$$

and covariance

$$\Sigma_{\vec{\beta}} = X^T R^{-1} X + \Sigma_0^{-1} \quad (14.220)$$

where as X^+ , R , and X^{+T} are invertible, one has

$$(X^+ R X^{+T})^{-1} = X^T R^{-1} X \quad (14.221)$$

Thus, by substitution to the original sample model, i.e., $\vec{z} = X^+ \vec{y}$, one has

$$f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y}) = f_N(\vec{\beta}; \hat{\vec{\beta}}, \Sigma_{\vec{\beta}}) \quad (14.222)$$

and

$$\hat{\vec{\beta}} = \Sigma_{\vec{\beta}} \Sigma_0^{-1} \vec{\mu}_0 + \Sigma_{\vec{\beta}} X^T R^{-1} \vec{y} \quad (14.223)$$

By the **matrix inversion lemma**, i.e.,

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}C)^{-1}DA^{-1} = A^{-1}B(C^{-1} + DA^{-1}C)^{-1}(BC)^{-1} \quad (14.224)$$

the covariance can be rewritten as

$$\Sigma_{\vec{\beta}} = (I - \Sigma_0 X^T (X \Sigma_0 X^T + R)^{-1} X) \Sigma_0 \quad (14.225)$$

or

$$\Sigma_{\vec{\beta}} = \Sigma_0 X^T (X \Sigma_0 X^T + R)^{-1} (X^T R^{-1})^{-1} \quad (14.226)$$

and the mean can be rewritten as

$$\hat{\vec{\beta}} = \left(I - \Sigma_0 X^T (X \Sigma_0 X^T + R)^{-1} X \right) \Sigma_0 \Sigma_0^{-1} \vec{\beta}_0 + \Sigma_0 X^T (X \Sigma_0 X^T + R)^{-1} (X^T R^{-1})^{-1} X^T R^{-1} \vec{y} \quad (14.227)$$

or

$$\hat{\vec{\beta}} = \vec{\beta}_0 - \Sigma_0 X^T (X \Sigma_0 X^T + R)^{-1} (\vec{y} - X \vec{\beta}_0) \quad (14.228)$$

which provides a recursive update to the prior mean, $\vec{\beta}_0$, through a gain on the residual, $\vec{y} - X \hat{\vec{\beta}}_0$. Thus, the **BLS parameter estimator** succinctly

$$\hat{\vec{\beta}}_{BLS} = \hat{\vec{\beta}}_0 + K (\vec{y} - X \hat{\vec{\beta}}_0) \quad (14.229)$$

where

$$K = \Sigma_0 X^T (X \Sigma_0 X^T + R)^{-1} \quad (14.230)$$

and

$$\Sigma_{\vec{\beta}} = (I - K X) \Sigma_0 \quad (14.231)$$

When R is unknown, one can specify a Gaussian-inverse-Wishart conjugate prior PDF, $\vec{\beta} \sim \mathcal{NIW}(\vec{\beta}, R; \vec{\mu}_0, \lambda_0, \Sigma_0, \nu_0)$ and consider a transformed sample as $\vec{z} = X^+ \vec{y}$ which produces a transformed regression model as

$$\vec{z} = \vec{\beta} + X^+ \vec{\epsilon} \quad (14.232)$$

and a transformed likelihood function given by

$$f_{\vec{Z}|\vec{B}_{\mu},\Sigma_B}(\vec{z}|\vec{\beta}, \beta_{\Sigma}) = f_{\mathcal{N}}(\vec{z}; \vec{\beta}_{\mu}, X^+ R X^{+T}) \quad (14.233)$$

with conjugate prior

$$f_{\vec{B},\Sigma_B}(\vec{\beta}, R) = f_{\mathcal{NIW}}(\vec{\beta}, R; \vec{\mu}_0, \lambda_0, \Sigma_0, \nu_0) \quad (14.234)$$

with corresponding transformed posterior PDF as

$$f_{\vec{B}_{\mu},\Sigma_B|\vec{Z}}(\vec{\beta}, R|\vec{z}) = f_{\mathcal{NIW}}(\vec{\beta}, R; \vec{\mu}, \lambda_0 + 1, \Phi, \nu_0 + 1) \quad (14.235)$$

with

$$\vec{\mu} = \frac{1}{\lambda_0 + 1} \vec{z} + \frac{\lambda_0}{\lambda_0 + 1} \vec{\mu}_0 \quad (14.236)$$

and

$$\Phi = \Phi_0 + \frac{\lambda_0}{\lambda_0 + 1} (\vec{z} - \vec{\mu}_0)(\vec{z} - \vec{\mu}_0)^T \quad (14.237)$$

Thus, by substitution to the original sample model, i.e., $\vec{z} = X^+ \vec{y}$, one has

$$f_{\vec{B}_{\mu},\Sigma_B|\vec{Y}}(\vec{\beta}, R|\vec{y}) = f_{\mathcal{NIW}}(\vec{\beta}, R; \vec{\mu}, \lambda_0 + 1, \Phi, \nu_0 + 1) \quad (14.238)$$

with mean

$$\vec{\mu} = \frac{1}{\lambda_0 + 1} (X^T X)^{-1} X \vec{y} + \frac{\lambda_0}{\lambda_0 + 1} \vec{\mu}_0 \quad (14.239)$$

and scale

$$\Phi = \Phi_0 + \frac{\lambda_0}{\lambda_0 + 1} ((X^T X)^{-1} X \vec{y} - \vec{\mu}_0)((X^T X)^{-1} X \vec{y} - \vec{\mu}_0)^T \quad (14.240)$$

Information Criterion for Multiple Model Selection

Mutual Information Criterion

Bayesian Information Criterion (BIC) and the **Widely Applicable Bayesian Information Criterion (WBIC)**

Akaike Information Criterion (AIC) and the **Widely Applicable Information Criterion (WAIC)**
Hannan-Quinn Information Criterion (HQC)

References

For more information, please refer to the following

- Gopalan, P., Hofman, J. M. and Blei, D. M., “Scalable recommendation with hierarchical Poisson factorization,” in *Proceedings of the Thirty-First Conference on Uncertainty in Artificial Intelligence (UAI’15)*, AUAI Press, 2015, pp. 326–335
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “3.3 Normal with a conjugate prior distribution,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, 2nd edition, pp. 67-69
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “3.5 Multivariate normal with known variance,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, 2nd edition, pp. 71-72
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “3.6 Multivariate normal with unknown mean and variance,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, pp. 72-74
- Sarkka, S., “2 Bayesian inference,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 17-26
- Sarkka, S., “3.2 Recursive linear regression,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 30-33
- Simon, D., “3.3 Recursive least squares estimation,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 131

14.5 Advanced Bayesian Optimal Parameter Estimation

If one is only interested in the MAP parameter estimate for the posterior, one of the simplest approximations may be to utilize an iterative search algorithm to find nearest mode or several modes, i.e., local maximum values, of the posterior PDF without the need to calculate the posterior PDF explicitly. In this case, one can utilize a mode-finding algorithm which attempts to maximize a quadratic approximation to the log posterior density, i.e.,

$$L(\vec{\beta} + \Delta\vec{\beta}) = \log f_{\vec{B}|\vec{Y}}(\vec{\beta} + \Delta\vec{\beta}|\vec{y}) \approx L(\vec{\beta}) + \vec{h}^T(\vec{\beta})\Delta\vec{\beta} + \frac{1}{2}\Delta\vec{\beta}^T H(\vec{\beta})\Delta\vec{\beta} \quad (14.241)$$

where $\vec{h}^T(\vec{\beta})$ and $H(\vec{\beta})$ are theoretically the gradient, $\nabla L(\vec{\beta})$ and the Hessian, $\frac{\partial^2 L(\vec{\beta})}{\partial\vec{\beta}\vec{\beta}^T}$, but may be approximations depending on the algorithm. The **Newton-Raphson algorithm** chooses $\vec{g}^T = \nabla L(\vec{\beta})$ and

$H = \frac{\partial L(\vec{\beta})}{\partial \vec{\beta}}$. Quasi-newton algorithms, e.g., the Broyden–Fletcher–Goldfarb–Shanno (BFGS) algorithm, approximate the Hessian. Finite difference methods approximate both the gradient and Hessian, e.g.,

$$g_i(\vec{\beta}) = \frac{L(\vec{\beta} + \delta_i \vec{1}_i) - L(\vec{\beta} - \delta_i \vec{1}_i)}{2\delta_i} \quad (14.242)$$

and

$$H_{i,j}(\vec{\beta}) = \frac{L(\vec{\beta} + \delta_i \vec{1}_i + \delta_j \vec{1}_j) - L(\vec{\beta} - \delta_i \vec{1}_i + \delta_j \vec{1}_j) - L(\vec{\beta} + \delta_i \vec{1}_i - \delta_j \vec{1}_j) + L(\vec{\beta} - \delta_i \vec{1}_i - \delta_j \vec{1}_j)}{4\delta_i\delta_j} \quad (14.243)$$

where $\delta_i \ll 1$ and $\vec{1}_i$ is the unit vector in the direction of $\vec{\beta}_i$.

The **mode-finding algorithm** can be written as

(1) Initialization:

- (a) Choose initial value $\hat{\vec{\beta}}(0)$
- (b) Choose convergence limit δ
- (c) Set $\frac{\|\vec{\Delta}(t)\|}{\|\hat{\vec{\beta}}(t)\|} < \infty$

(2) While $\frac{\|\vec{\Delta}(t)\|}{\|\hat{\vec{\beta}}(t)\|} > \delta$ and $t \leq T$:

- (a) Compute A and B
- (b) Compute $B(\vec{\beta})$
- (c) Set $\hat{\vec{\beta}}(t+1) = \hat{\vec{\beta}}(t) - B^{-1}(\vec{\beta}) \vec{a}(\vec{\beta})$
- (d) Set $t = t + 1$

(3) Select mode of $f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y})$ as

$$\hat{\vec{\beta}}_{mode} = \hat{\vec{\beta}}(t) \quad (14.244)$$

Repeating this mode-finding algorithm, one may generally find K modes and approximate the MAP parameter estimator as

$$\hat{\vec{\beta}}_{MAP} \approx \max_{t=1,\dots,K} \hat{\vec{\beta}}_{mode}(t) \quad (14.245)$$

After estimating K modes, one can also approximate the posterior distribution itself by a finite mixture of distributions based on each mode. The most common is the **Gaussian-mixture (GM) approximation**

$$f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y}) = \sum_{k=1}^K w_k f_N(\vec{\beta}; \hat{\vec{\beta}}_{mode}(k), V_{\vec{\beta}}) \quad (14.246)$$

where

$$V_{\vec{\beta}} = \left(-\frac{\partial^2 L(\vec{\beta})}{\partial \vec{\beta} \vec{\beta}^T} \right)^{-1} \quad (14.247)$$

and enforcing

$$\sum_{k=1}^K w_k = 1 \quad (14.248)$$

The weights, w_k , are estimated based on matching $f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y})$ or its un-normalized approximation at all $\vec{\beta}(k)$. An alternative for heavy-tailed distributions is the t -mixture approximation.

The mixture approximation may not be appropriate for all elements of the parameter vector. Thus, one can supplement the mixture approximation approach with an expectation-maximization (EM) algorithm which iterates between an expectation step (E-step) and a maximization step (M-step). This approach notably utilizes a partitioning of the parameter vector, $\vec{\beta}$, into two sub-parameter vectors, a modeled $\vec{\gamma}$ and an unmodeled $\vec{\phi}$, i.e.,

$$\vec{\beta} = \begin{bmatrix} \vec{\gamma} \\ \vec{\phi} \end{bmatrix} \quad (14.249)$$

such that the analytical approximation works well for $f_{\vec{\Gamma}, \vec{\Phi}|\vec{Y}}(\vec{\gamma}, \vec{\phi}|\vec{y}, \vec{y})$. Now consider the joint distribution of $\vec{\beta}$ as

$$f_{\vec{\Gamma}, \vec{\Phi}|\vec{Y}}(\vec{\gamma}, \vec{\phi}|\vec{y}) = f_{\vec{\Gamma}|\vec{\Phi}, \vec{Y}}(\vec{\gamma}|\vec{\phi}, \vec{y}) f_{\vec{\Phi}|\vec{Y}}(\vec{\phi}|\vec{y}) \quad (14.250)$$

Next, taking the logarithm of both sides and rearranging, one has

$$\log f_{\vec{\Phi}|\vec{Y}}(\vec{\phi}|\vec{y}) = f_{\vec{\Gamma}, \vec{\Phi}|\vec{Y}}(\vec{\gamma}, \vec{\phi}|\vec{y}) - \log f_{\vec{\Gamma}|\vec{\Phi}, \vec{Y}}(\vec{\gamma}|\vec{\phi}, \vec{y}) \quad (14.251)$$

Then, by taking the expectation with respect to the sub-parameter vector $\vec{\gamma}$ under the distribution $f_{\vec{\Gamma}|\vec{\Phi}(t), \vec{Y}}(\vec{\gamma}|\vec{\phi}(t), \vec{y})$, i.e., averaging over all values of $\vec{\gamma}$, one has

$$\log f_{\vec{\Phi}|\vec{Y}}(\vec{\phi}|\vec{y}) = \mathbb{E}_t \left[\log f_{\vec{\Gamma}, \vec{\Phi}|\vec{Y}}(\vec{\gamma}, \vec{\phi}|\vec{y}) \right] - \mathbb{E} \left[\log f_{\vec{\Gamma}|\vec{\Phi}, \vec{Y}}(\vec{\gamma}|\vec{\phi}, \vec{y}) \right] \quad (14.252)$$

where the second term on the right side is maximum at $\vec{\phi}(t)$ and the first term on the right side could be maximized for a new $\vec{\phi}(t+1)$. This would increase the marginal PDF $f_{\vec{\Phi}|\vec{Y}}(\vec{\phi}|\vec{y})$ at each iteration which guarantees the EM algorithm will converge to a local mode of the posterior PDF except in some special cases.

The **expectation-maximization (EM) algorithm** can be written as

(1) Initialization:

- (a) Choose initial value $\hat{\vec{\phi}}(0)$
- (b) Choose convergence limit δ
- (c) Choose total number of iterations, T

(2) While $\frac{\|\hat{\vec{\phi}}(t) - \hat{\vec{\phi}}(t-1)\|}{\|\hat{\vec{\phi}}(t)\|} > \delta$ and $t \leq T$:

- (a) *E-step*: Compute

$$\mathbb{E}_t \left[\log f_{\vec{\Gamma}, \vec{\Phi}|\vec{Y}}(\vec{\gamma}, \vec{\phi}|\vec{y}) \right] = \int \cdots \int \log \left(f_{\vec{\Gamma}, \vec{\Phi}|\vec{Y}}(\vec{\gamma}, \vec{\phi}|\vec{y}) \right) f_{\vec{\Gamma}|\vec{\Phi}, \vec{Y}}(\vec{\gamma}|\vec{\phi}(t), \vec{y}) d\vec{\gamma} \quad (14.253)$$

(b) *M-step:* Compute

$$\hat{\vec{\phi}}(t+1) = \underset{\hat{\vec{\phi}}}{\operatorname{argmax}} \mathbb{E} \left[\log f_{\vec{\Gamma}, \vec{\Phi} | \vec{Y}}(\vec{\gamma}, \vec{\phi} | \vec{y}) \right] \quad (14.254)$$

(c) Set $t = t + 1$

(3) Select EM mode estimates as

$$\hat{\vec{\phi}}_{mode} = \hat{\vec{\phi}}(t) \quad (14.255)$$

Repeating this mode-finding algorithm, one may generally find K modes and approximate the MAP parameter estimator as

$$\hat{\vec{\phi}}_{MAP} \approx \max_{t=1, \dots, K} \hat{\vec{\phi}}_{mode}(t) \quad (14.256)$$

The “hyper-parameters,” $\vec{\phi}$, maximization of EM as a point estimate of the MAP can be extended to **variational Bayes** methods which analytically approximate the posterior PDF by an alternate parameterized variational distribution, $\bar{q}(\vec{\beta} | \vec{\phi})$, with selected hyper-parameters, $\vec{\phi}$. This approximation is iteratively chosen such the the K-L divergence is minimized, i.e.,

$$\vec{\phi}_{opt} = \underset{\vec{\phi}}{\operatorname{argmin}} KL(\bar{q} || f_{\vec{B} | \vec{Y}}) = \underset{\vec{\phi}}{\operatorname{argmin}} - \int \cdots \int \bar{q}(\vec{\beta} | \vec{\phi}) \log \left(\frac{f_{\vec{B} | \vec{Y}}(\vec{\beta} | \vec{y})}{\bar{q}(\vec{\beta} | \vec{\phi})} \right) d\vec{\beta} \quad (14.257)$$

Then, the **VB parameter estimator** is given by

$$\hat{\vec{\beta}}_{VB} = \int \cdots \int g(\vec{\beta}) \bar{q}(\vec{\beta} | \vec{\phi}) d\vec{\beta} \quad (14.258)$$

For tractability, one typically assumes the variational distribution has M independent parameters, i.e.,

$$\bar{q}(\vec{\beta} | \vec{\phi}) = \prod_{j=1}^M \bar{q}_j(\beta_j | \phi_j) \quad (14.259)$$

and

$$\bar{q}_{-j}(\vec{\beta}_{-j} | \vec{\phi}_{-j}) = \bar{q}_1(\beta_1 | \phi_1) \cdots \bar{q}_{j-1}(\beta_{j-1} | \phi_{j-1}) \bar{q}_{j+1}(\beta_{j+1} | \phi_{j+1}) \cdots \bar{q}_M(\beta_M | \phi_M) \quad (14.260)$$

Then, one iteration of variational Bayes approximation updates the hyper-parameters, $\vec{\phi}$, so that

$$\log(\bar{q}_j(\beta_j | \phi_j)) = \int \log(f_{\vec{B} | \vec{Y}}(\vec{\beta} | \vec{y})) \bar{q}_{-j}(\vec{\beta}_{-j} | \vec{\phi}_{-j}) d\vec{\beta}_{-j} \quad (14.261)$$

which is similar to Gibbs sampling shown later except the the update uses the average over the other parameters instead of the conditional PDF. This method can be shown by calculus of variations, hence the name, that this iteration converges to $\vec{\phi}_{opt}$. Because of the use of the logarithm of the posterior, variational Bayes works best on exponential family models with conditionally conjugate prior distributions in which case approximating variational distribution can be determined by inspection and the necessary expectations performed in closed-form.

Numerical Integration for Parameter Estimation

In many cases, no analytical solutions exist for computing the integrals for the Bayes posterior, its modes, and/or the Bayes estimator using the expectation integral of the posterior, e.g.,

$$\hat{\vec{\beta}}_{BE} = \mathbb{E}_{\vec{\beta}|\vec{Y}}[g(\vec{\beta})] = \int \cdots \int g(\vec{\beta}) f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y}) d\vec{\beta} \quad (14.262)$$

where $g(\vec{\beta})$ is some chosen statistic of the posterior. One approach is to approximate the posterior by an analytical approximation that provides closed-form integration, e.g., Laplace's method, or to optimally fit the posterior to a model, e.g., Variational Bayes. An alternative is to employ **numerical integration**, also known as **quadrature** methods, in which the integral is evaluated at discrete points. Numerical integration methods are divided between deterministic quadrature rule methods or stochastic Monte Carlo methods.

Quadrature rule (QR) methods numerically approximate the expectation integral by a weighted sum of M specified **integration-points** in the parameter space which forms the **QR parameter estimator** by

$$\hat{\vec{\beta}}_{QR} \approx \operatorname{argmin}_{\vec{\beta}} \sum_{m=1}^M w_m g(\vec{\beta}(m)) \quad (14.263)$$

where $\vec{\beta}(m)$ is the m^{th} integration-point with corresponding weight, w_m . The choice of weights that provide a good approximation are intrinsically related to the posterior, $f_{\vec{\beta}(m)|\vec{Y}}(\vec{\beta}(m)|\vec{y})$, which may be unknown. A typical use of QR methods is to use an iterative approach to selecting the weights or to assume a form for $f_{\vec{\beta}(m)|\vec{Y}}(\vec{\beta}(m)|\vec{y})$, e.g., a multivariate Gaussian with mean $\vec{\mu}$ and covariance Σ . In the Gaussian case, one may use transform the integration-points to unit **sigma-points**, $\vec{\xi}(m)$, and corresponding weights w_m through **stochastic decoupling**, i.e.

$$\hat{\vec{\beta}}_{SP} \approx \operatorname{argmin}_{\vec{\beta}} \sum_{m=1}^M w_m g(\vec{\mu} + \Sigma^{1/2} \vec{\xi}(m)) \quad (14.264)$$

There are different methods for constructing these weights and sigma-points, e.g., the central difference, unscented transform, cubature transform, Gauss-Hermite rule, Gaussian process quadratures, etc. These methods are discussed in a following chapter on nonlinear Bayesian state estimation for which sigma-points are considered as different statistical linear regressions of statistical linearization.

Monte Carlo (MC) methods numerically approximate the expectation integral by a weighted sum of M IID samples of $\vec{\beta} \sim f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y})$, i.e.,

$$\hat{\vec{\beta}}_{MC} \approx \operatorname{argmin}_{\vec{\beta}} \frac{1}{M} \sum_{m=1}^M \mathcal{T}(\vec{\beta}(m), \hat{\vec{\beta}}) \quad (14.265)$$

where $\vec{\beta}(m)$ is the m^{th} sample. By the law of large numbers, one can show that $\hat{\vec{\beta}}^{MC} \rightarrow \hat{\vec{\beta}}_{BE}$ as $M \rightarrow \infty$. Unfortunately, it is also typically the case that one cannot draw samples directly from $f_{\vec{\beta}|\vec{Y}}(\vec{\beta}|\vec{y})$, so one must perform point-wise evaluations of the **target function**

$$\pi(\vec{\beta}) = f_{\vec{Y}|\vec{\beta}}(\vec{y}|\vec{\beta}) f_{\vec{\beta}}(\vec{\beta}) \quad (14.266)$$

which is notably the numerator of Bayes' Rule for the posterior PDF, but is not a proper PDF.

These MC methods all fall under three categories: rejection sampling, Markov Chain, and importance sampling, which will be summarized here. In all three cases, one uses one or more **proposal functions**, $q(\vec{\beta})$, or the related **proposal PDF**, $\bar{q}(\vec{\beta})$, where

$$\bar{q}(\vec{\beta}) = \frac{q(\vec{\beta})}{f_{\vec{Y}}(\vec{y})} \quad (14.267)$$

to generate samples from any posterior PDF, $f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y})$.

Rejection sampling Monte Carlo (RSMC) methods draw samples from a proposal PDF, $q(\vec{\beta})$, instead of $f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y})$, then the sample is either accepted or rejected based on a ratio of the two PDFs from which it can be proven that the accepted samples are then distributed according the posterior PDF. For such ratios, one must use an envelope function, i.e. $Cq(\vec{\beta}) \geq \pi(\vec{\beta})$ for all possible $\vec{\beta}$ where C is some constant. However, it should be noted that finding such a C for arbitrary $f_{\vec{B}|\vec{Y}}(\vec{\beta}|\vec{y})$ may be difficult and the acceptance rate may be small for a large portion of the parameter space, and the number of iterations required for completion is not known *a priori* and make the RSMC very inefficient.

The RSMC algorithm can be written as

(1) Initialization:

- (a) Choose proposal function $q(\vec{\beta})$
- (b) Choose required number of samples for posterior estimate: M
- (c) Find upper bound, $C \geq \frac{\pi(\vec{\beta})}{q(\vec{\beta})}$ for all possible $\vec{\beta}$
- (d) Let $t = m = 1$

(2) While $m < M$:

- (a) Draw sample from $\vec{\beta}(t) \sim \bar{q}(\vec{\beta})$
- (b) Draw sample from $u \sim \mathcal{U}(0, 1)$
- (c) If $u \leq \frac{\pi(\vec{\beta}(t))}{Cq(\vec{\beta})(t)}$
 - (i) Set $\vec{\beta}(m) = \vec{\beta}(t)$
 - (ii) Set $m = m + 1$
- (d) Set $t = t + 1$, regardless of (c)

(3) Compute the **RSMC parameter estimator** by

<3->

$$\hat{\vec{\beta}}^{RSMC} = \frac{1}{M} \sum_{m=1}^M g(\vec{\beta}(m)) \quad (14.268)$$

Markov Chain Monte Carlo (MCMC) methods produce an Markov Chain whose stationary density is the desired posterior PDF, $f_{\vec{\beta}|\vec{y}}(\vec{\beta}|\vec{y})$, of which the most common are the Gibbs sampler and the Metropolis-Hastings algorithm (MHA). The **Gibbs sampler**, also known as **alternate conditional sampling**, which assumes one can choose $M \leq n_\beta$ sub-vectors of $\vec{\beta}$ such that one has well-defined conditional posterior PDFs for the individual sub-vectors of $\vec{\beta}$, i.e., for $j = 1, \dots, M$,

$$q_j(\vec{\beta}) = f_{\vec{\beta}_j|\vec{\beta}_{-j}(t-1), \vec{y}}(\vec{\beta}_j|\vec{\beta}_{-j}(t-1), \vec{y}) \quad (14.269)$$

where

$$\vec{\beta}_{-j}(t-1) = \begin{bmatrix} \vec{\beta}_1(t) \\ \vdots \\ \vec{\beta}_{j-1}(t-1) \\ \vec{\beta}_{j+1}(t) \\ \vdots \\ \vec{\beta}_M(t-1) \end{bmatrix} \quad (14.270)$$

Then, by performing M sampling steps for each iteration t of the sampling process, it can be shown that one will converge to the sampling from $f_{\vec{\beta}|\vec{y}}(\vec{\beta}|\vec{y})$ as $t \rightarrow \infty$. This is especially useful for known mixture distributions as the true posterior distribution.

The MHA can be considered as a generalized rejection sampler whose proposal PDF depends on the result of the previous iteration, i.e., on $\vec{\beta}(t-1)$. Moreover, as the acceptance rate depends on $\vec{\beta}(t-1)$, the value is re-used whenever a candidate sample is rejected, thus the samples are no longer IID, but as the underlying Markov Chain has the desired posterior as its limiting stationary distribution, this resampling is done with the right acceptance probability to ensure its validity. The Gibbs sampler can also be seen as a special case of the MHA where every sample is accepted due to the construction of the proposal function via M steps of sampling the M subvectors at each iteration, t .

The MHA-MCMC parameter estimator can be outlined as

(1) Initialization:

- (a) Choose proposal function $q(\vec{\beta}|\vec{\beta}(t-1))$
- (b) Choose initial state $\vec{\beta}(0)$
- (c) Choose total number of iterations, T
- (d) Choose burn-in period, T_b

(2) For $t = 1, \dots, T$:

- (a) Draw sample from $\vec{\beta}^* \sim q(\vec{\beta}|\vec{\beta}(t-1))$
- (b) Draw sample from $u \sim \mathcal{U}(0, 1)$
- (c) Compute acceptance probability:

$$\alpha(\vec{\beta}^*, \vec{\beta}(t)) = \min \left[1, \frac{\pi(\vec{\beta}^*) q(\vec{\beta}(t-1)|\vec{\beta}^*)}{\pi(\vec{\beta}(t-1)) q(\vec{\beta}^*|\vec{\beta}(t-1))} \right] \quad (14.271)$$

- (d) If $u \leq \alpha(\vec{\beta}(t), \vec{\beta}(t-1))$
 - (i) Set $\vec{\beta}(t) = \vec{\beta}^*$
 - (e) Otherwise
 - (i) Set $\vec{\beta}(t) = \vec{\beta}(t-1)$
- (3) Compute the **MHA-MCMC parameter estimator** by

<3->

$$\hat{\vec{\beta}}_{MHA-MCMC} = \frac{1}{T - T_b} \sum_{t=T_b+1}^T g(\vec{\beta}(t)) \quad (14.272)$$

Importance Sampling Monte Carlo (ISMC) methods draw samples from one or more proposal PDFs, $\bar{q}(\vec{\beta})$, however, ISMC methods accept all samples, but assign each sample a weight according to their quality in approximating the desired posterior PDF. The primary division between methods being the single or multiple proposals used in ISMC method. ISMC methods may also be non-adaptive or adaptive where the parameters of proposal(s) are adapted iteratively to better approximate the desired posterior PDF.

Thus, the standard, non-adaptive ISMC algorithm can be written as

- (1) Choose single proposal function $\bar{q}(\vec{\beta})$
- (2) Draw M samples from $\vec{\beta}(m) \sim \bar{q}(\vec{\beta})$ for $m = 1, \dots, M$
- (3) Compute M importance weights:

$$w(m) = \frac{\pi(\vec{\beta}(m))}{\bar{q}(\vec{\beta})(m)} \quad (14.273)$$

- (4) Estimate marginal likelihood by

$$\hat{f}_{\vec{Y}} = \frac{1}{M} \sum_{m=1}^M w(m) \quad (14.274)$$

- (5) Compute the **ISMC parameter estimator** by

$$\hat{\vec{\beta}}_{ISM} = \frac{1}{M \hat{f}_{\vec{Y}}} \sum_{m=1}^M w(m) g(\vec{\beta}(m)) \quad (14.275)$$

References

For more information, please refer to the following

- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “10.1 Numerical integration,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, pp. 72-74
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “10.3 Direct simulation and rejection sampling,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, pp. 263-265
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “10.4 Importance sampling,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, pp. 265-267
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “11.1 Gibbs sampler,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, pp. 276-278
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “11.2 Metropolis and Metropolis-Hastings algorithms,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, pp. 278-280
- Gelman, A., Carlin, J., Stern, H., and Rubin, D., “13.7 Variational inference,” in *Bayesian Data Analysis*, 3rd Ed., Chapman and Hall, 2004, pp. 331-338
- Sarkka, S., “7.1 Monte Carlo approximations in Bayesian inference,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 116-117
- Sarkka, S., “7.2 Importance Sampling,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 117-120

Optimal Linear State Estimation Theory

15.1 Introduction to Optimal State Estimation

Decision theory is the determination of the optimal decision given constraints and assumptions on a decision problem. **State estimation** is one special case of a model-based decision problem and can be stated as deciding which state, \vec{x} , one should select as a function of time. However, the Bayesian state estimation problem assumes that the state, \vec{x} , is “hidden” and only observed through measurements, \vec{y} , through which Bayesian inference must be performed. With the inherent time-dependency of dynamical state models, three different types of time horizons can be considered for the type of state estimation problem and the type of state estimator. State estimation for a *future* state is known as **predicting** which utilizes a mathematical law or algorithm known as a **state predictor**. State estimation for a *current* state is known as **state filtering** which utilizes a mathematical law or algorithm known as a **state filter**. State estimation for a *past* state is known as **state smoothing** which utilizes a mathematical law or algorithm known as a **state smoother**.

In addition to this general problem for state estimation, perception systems use proprioceptive and exteroceptive sensors to observe the system and infer the state through Bayes state estimation using a model of the sensor with respect to the state and/or the state change. As every sensor exhibits some degree of inaccuracy, uncertainty, or random phenomena in its measurements, one must additionally model sensor **noise** as opposed to the modeled **signal** which are two terms based in signal processing theory. Therefore, as these noises are unknown, i.e. one cannot predict with absolute certainty what their exact values will be in the future, one instead models the noise as stochastic processes. Another perspective on the uncertainty modeling is due to the fact that with the stochastic dynamical system model, one is assuming certain aspects about the modeled process selected to represent the actual system, environment, and sensors. Any tractable sensor model cannot be exactly quantified at every instant, e.g., calibration errors or the electro-mechanical variations within an individual sensor. These sensor model errors can generally be dynamic uncertainties and contribute to the random nature of sensors.

Bayesian State Estimation

With these concepts in mind, one typically utilizes a Bayesian approach to model-based state estimation which will be presented here for general discrete-time state and measurement process models where the subscripts denote the time step indices of the vectors with the “ $1 : N$ ” denoting grouping together all vectors with indices as $1, \dots, 5$. Thus, the complete state and measurement joint PDF is

$$f_{\vec{X}_{0:N}, \vec{Y}_{1:N}}(\vec{x}_{0:N}, \vec{y}_{1:N}) = f_{\vec{X}_{0:N} | \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{y}_{1:N}) f_{\vec{Y}_{1:N}}(\vec{y}_{1:N}) \quad (15.1)$$

where “joint” denotes all time steps, $k = 1, \dots, N$. This can be decomposed theoretically by Bayes’ rule in the joint state posterior PDF

$$f_{\vec{X}_{0:N} | \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{y}_{1:N}) = \frac{f_{\vec{Y}_{1:N} | \vec{X}_{0:N}}(\vec{y}_{1:N} | \vec{x}_{0:N}) f_{\vec{X}_{0:N}}(\vec{x}_{0:N})}{f_{\vec{Y}_{1:N}}(\vec{y}_{1:N})} \quad (15.2)$$

where $f_{\vec{X}_{0:N}}(\vec{x}_{0:N})$ is the prior distribution including the previous time step given by the state process/dynamics model, $f_{\vec{Y}_{1:N} | \vec{X}_{0:N}}(\vec{y}_{1:N} | \vec{x}_{0:N})$ is the likelihood model of the measurement process, and $f_{\vec{X}_{0:N} | \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{y}_{1:N})$ is the joint state posterior PDF given the observed measurement history. Recalling the discussion for Bayesian parameter estimation, if one computes the state posterior PDF, one can use different optimality criteria for the **Bayes state estimator**, e.g., the **maximum *a posteriori* (MAP)** as the maximal mode of the posterior, the **minimum mean square error (MMSE)** as the mean of the posterior for finite means and variances, and the **median estimator** as the median of the posterior.

However, this general Bayes formulation is difficult to compute useful information especially if new measurements are obtained in time k and the state should be dynamically estimated as a function of k , e.g., in a perception system. Thus, for computational tractability, one can simplify the Bayes state estimation problem with the **hidden Markov model (HMM) assumption** which assumes the state process follow an n_x^{th} -order Markov process, i.e., $f_{\vec{X}}(\vec{x}_k | \vec{x}_{k-1})$, which allows one to model the joint state prior PDF as

$$f_{\vec{X}_{0:N}}(\vec{x}_{0:N}) = \left(\prod_{k=1}^N f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k | \vec{x}_{k-1}) \right) f_{\vec{X}_0}(\vec{x}_0) \quad (15.3)$$

where $f_{\vec{X}}(\vec{x}_0)$ is the **state prior PDF** at the initial time step $k = 0$ and $f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k | \vec{x}_{k-1})$ is the **state transition PDF** from time step $k - 1$ to k . Furthermore, as the measurement of the Markov state at time k is conditionally independent of other states and measurements, i.e., $f_{\vec{Y}}(\vec{y}_k | \vec{x}_k)$ is the measurement model at k that depends only on the state at that time k , one has the joint likelihood between of the state and measurements as

$$f_{\vec{Y}_{1:N} | \vec{X}_{0:N}}(\vec{y}_{1:N} | \vec{x}_{0:N}) = \prod_{k=1}^N f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k) \quad (15.4)$$

With the HMM assumption, the joint state posterior PDF can be rewritten as

$$f_{\vec{X}_{0:N} | \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{y}_{1:N}) = \frac{\left(\prod_{k=1}^N f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k) f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k | \vec{x}_{k-1}) \right) f_{\vec{X}}(\vec{x}_0)}{f_{\vec{Y}_{1:N}}(\vec{y}_{1:N})} \quad (15.5)$$

which could be solved similar to batch parameter estimators as stacked state trajectory over all time steps could be considered as single “parameter vector” to be estimated. However, computing the complete joint

posterior distribution is often unnecessary and inefficient for real-time perception as the number of samples will change with time. Thus, one is typically interested in one of three different posteriors of the complete joint PDF of the state and measurement, namely, the **state prediction posterior PDF**

$$f_{\vec{X}_{k+n} | \vec{Y}_{1:k}}(\vec{x}_{k+n} | \vec{y}_{1:k}) \quad (15.6)$$

with $n > 0$ constant, the **state filtering posterior PDF**

$$f_{\vec{X}_k | \vec{Y}_{1:k}}(\vec{x}_k | \vec{y}_{1:k}) \quad (15.7)$$

and the **state smoothing posterior PDF**

$$f_{\vec{X}_k | \vec{Y}_{1:k+n}}(\vec{x}_k | \vec{y}_{1:k+n}) \quad (15.8)$$

where if $0 < n < N - k$ and constant, one has a **fixed-lag state smoothing posterior PDF**, and if $n = N - k$ for all k with N constant, one has a **fixed-interval state smoothing posterior**. Thus, fixed-interval smoothers can sometimes be considered the same as batch parameter estimators as the stacked state trajectory over all time steps could be considered as a single “parameter vector” to be estimated.

Recall that if with the HMM assumption, one can alternatively represent the state process, $f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k | \vec{x}_{k-1})$, and the measurement likelihood, $f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k)$, with the discrete-time stochastic state-space model

$$\begin{aligned} \vec{x}[k+1] &= f(\vec{x}[k], \vec{u}[k], \vec{w}[k], k) \\ \vec{y}[k] &= h(\vec{x}[k], \vec{v}[k], k) \end{aligned} \quad (15.9)$$

where $\vec{w}[k]$ and $\vec{v}[k]$ represent the process and measurement noise with some known distribution related to $f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k | \vec{x}_{k-1})$ and $f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k)$.

Furthermore, one can also consider an equivalent notation and development of the HMM for continuous-time and hybrid-time Bayesian state estimation which allows similar stochastic state-space models, i.e., for continuous-time as

$$\begin{aligned} \dot{\vec{x}}(t) &= f(\vec{x}(t), \vec{u}(t), \vec{w}(t), t) \\ \vec{y}(t) &= h(\vec{x}(t), \vec{v}(t), t) \end{aligned} \quad (15.10)$$

and hybrid-time as

$$\begin{aligned} \dot{\vec{x}}(t) &= f(\vec{x}(t), \vec{u}(t), \vec{w}(t), t) \\ \vec{y}[k] &= h(\vec{x}[k], \vec{v}[k], k) \end{aligned} \quad (15.11)$$

Bayes Predictor

The **Bayes predictor** estimates the state posterior PDF at time step $k + n$ assuming one has knowledge of the state posterior PDF at some time step k , i.e.,

$$f_{\vec{X}_0 | \vec{Y}_{1:0}}(\vec{x}_0 | \vec{y}_{1:0}) \quad (15.12)$$

which for $k = 0$, one has $f_{\vec{X}_k|\vec{Y}_{1:0}}(\vec{x}_k|\vec{y}_{1:0}) = f_{\vec{X}}(\vec{x}_0)$. Otherwise, this could be computed using the Bayes filter or smoother equations below. Then, note that the joint state posterior PDF between future state $\vec{X}_{k+n}, \dots, \vec{x}_{k+1}$ and the current state \vec{x}_k can be defined as

$$f_{\vec{X}_{k:k+n}|\vec{Y}_{1:k}}(\vec{x}_{k+n}, \dots, \vec{x}_k|\vec{y}_{1:k}) = \left(\prod_{i=0}^{n-1} f_{\vec{X}_{k+i+1}|\vec{X}_{k+i}, \vec{Y}_{1:k}}(\vec{x}_{k+i+1}|\vec{x}_{k+i}, \vec{y}_{1:k}) \right) f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) \quad (15.13)$$

and by the Markov property for the state process, one has

$$f_{\vec{X}_{k:k+n}|\vec{Y}_{1:k}}(\vec{x}_{k+n}, \dots, \vec{x}_k|\vec{y}_{1:k}) = \left(\prod_{i=0}^{n-1} f_{\vec{X}_{k+i+1}|\vec{X}_{k+i}}(\vec{x}_{k+i+1}|\vec{x}_{k+i}) \right) f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) \quad (15.14)$$

Thus, the **state prediction posterior PDF** at $k+n$ can be computed by marginalizing out $\vec{X}_k, \dots, \vec{X}_{k+n-1}$, i.e.

$$f_{\vec{X}}(\vec{x}_{k+n}|\vec{y}_{1:k}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \left(\prod_{i=0}^{n-1} f_{\vec{X}_{k+i+1}|\vec{X}_{k+i}}(\vec{x}_{k+i+1}|\vec{x}_{k+i}) \right) f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) d\vec{x}_k \cdots d\vec{x}_{k+n-1} \quad (15.15)$$

For $n = 1$, one obtains the **Chapman-Kolmogorov equation** for one-step Bayes state prediction as

$$f_{\vec{X}_{k+1}|\vec{Y}_{1:k}}(\vec{x}_{k+1}|\vec{y}_{1:k}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{X}_{k+1}|\vec{X}_{k+1}}(\vec{x}_{k+1}|\vec{x}_k) f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) d\vec{x}_k \quad (15.16)$$

which can be implemented recursively to predict the state forward in time for any $k+1, k+2, \dots$. This can also be seen by the independence of the transition PDFs in the product which can be pulled out into individual sets of integrals over $\vec{x}_k, \vec{x}_{k+1}, \dots$

Bayes Filter

The **Bayes filter**, also known as the **recursive Bayes estimator**, updates or “corrects” the state prediction PDF with the currently available measurement to produce the state filtering posterior PDF. This prediction/correction framework results in two separate state filtering steps, the **prediction** and **correction** steps. This recursion assumes one has knowledge of the state filtering posterior PDF at the previous time step $k-1$, i.e.,

$$f_{\vec{X}_{k-1}|\vec{Y}_{1:k-1}}(\vec{x}_{k-1}|\vec{y}_{1:k-1}) \quad (15.17)$$

which for $k = 1$, one has $f_{\vec{X}_0|\vec{Y}_{1:0}}(\vec{x}_0|\vec{y}_{1:0}) = f_{\vec{X}}(\vec{x}_0)$. The **prediction step** uses the Chapman-Kolmogorov equation as

$$f_{\vec{X}_k|\vec{Y}_{1:k-1}}(\vec{x}_k|\vec{y}_{1:k-1}) = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} f_{\vec{X}_{k+1}|\vec{X}_{k+1}}(\vec{x}_{k+1}|\vec{x}_k) f_{\vec{X}_{k-1}|\vec{Y}_{1:k-1}}(\vec{x}_{k-1}|\vec{y}_{1:k-1}) d\vec{x}_k \quad (15.18)$$

where $f_{\vec{X}_k}(\vec{x}_k|\vec{y}_{1:k-1})$ is the **state filtering prior PDF**, i.e., the state conditional PDF “prior”-measurement at k .

Recalling Bayes rule’ for the state filtering posterior PDF as

$$f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) = \frac{f_{\vec{Y}_k|\vec{X}_k, \vec{Y}_{1:k-1}}(\vec{y}_k|\vec{x}_k, \vec{y}_{1:k-1}) f_{\vec{X}_k|\vec{Y}_{1:k-1}}(\vec{x}_k|\vec{y}_{1:k-1})}{\int \cdots \int f_{\vec{Y}_k|\vec{X}_k, \vec{Y}_{1:k-1}}(\vec{y}_k|\vec{x}_k, \vec{y}_{1:k-1}) f_{\vec{X}_k|\vec{Y}_{1:k-1}}(\vec{x}_k|\vec{y}_{1:k-1}) d\vec{x}_k} \quad (15.19)$$

which, by the conditional independence of likelihoods, the state filtering prior PDF is updated with the **correction step**

$$f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) = \frac{f_{\vec{Y}_k|\vec{X}_k}(\vec{y}_k|\vec{x}_k)f_{\vec{X}_k|\vec{Y}_{1:k-1}}(\vec{x}_k|\vec{y}_{1:k-1})}{\int \cdots \int f_{\vec{Y}_k|\vec{X}_k}(\vec{y}_k|\vec{x}_k)f_{\vec{X}_k|\vec{Y}_{1:k-1}}(\vec{x}_k|\vec{y}_{1:k-1})d\vec{x}_k} \quad (15.20)$$

where $f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k})$ is the **state filtering posterior PDF**, i.e., the state conditional PDF “post”-measurement at k .

Fixed-Interval Bayes Smoother

The **fixed-interval Bayes smoother**, also known as the **backwards recursive Bayes estimator**, updates or “smooths” the state filtering prior and posterior PDFs with future measurements up to time step N to produce the fixed-interval state smoothing posterior PDF. This prediction-correction-smoothing framework results in three separate state smoothing steps. First, one computes the state filtering prior PDFs for all $k = 0, \dots, N$

$$f_{\vec{X}_{k+1}|\vec{Y}_{1:k}}(\vec{x}_{k+1}|\vec{y}_{1:k}) \quad (15.21)$$

and the state filtering posterior PDFs for all $k = 0, \dots, N$

$$f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) \quad (15.22)$$

which for $k = 0$, the state filtering posterior PDF is the state prior PDF and for $k = N$, the state filtering posterior PDF is the state smoothing posterior PDF at N .

The joint two-state posterior PDF for \vec{X}_k and \vec{X}_{k+1} can be written as

$$f_{\vec{X}_k, \vec{X}_{k+1}|\vec{Y}_{1:k}}(\vec{x}_k, \vec{x}_{k+1}|\vec{y}_{1:k}) = f_{\vec{X}_k|\vec{X}_{k+1}, \vec{Y}_{1:N}}(\vec{x}_k|\vec{x}_{k+1}, \vec{y}_{1:N})f_{\vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_{k+1}|\vec{y}_{1:N}) \quad (15.23)$$

and by the conditional independence of the likelihoods, one has

$$f_{\vec{X}_k, \vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_k, \vec{x}_{k+1}|\vec{y}_{1:N}) = f_{\vec{X}_k|\vec{X}_{k+1}, \vec{Y}_{1:k}}(\vec{x}_k|\vec{x}_{k+1}, \vec{y}_{1:k})f_{\vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_{k+1}|\vec{y}_{1:N}) \quad (15.24)$$

By Bayes’ rule for the first term, one has

$$f_{\vec{X}_k, \vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_k, \vec{x}_{k+1}|\vec{y}_{1:N}) = \frac{f_{\vec{X}_{k+1}|\vec{X}_k, \vec{Y}_{1:k}}(\vec{x}_{k+1}|\vec{x}_k, \vec{y}_{1:k})f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k})}{f_{\vec{X}_{k+1}|\vec{Y}_{1:k}}(\vec{x}_{k+1}|\vec{y}_{1:k})}f_{\vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_{k+1}|\vec{y}_{1:N}) \quad (15.25)$$

By the HMM assumption, one has

$$f_{\vec{X}_k, \vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_k, \vec{x}_{k+1}|\vec{y}_{1:N}) = \frac{f_{\vec{X}_{k+1}|\vec{X}_k}(\vec{x}_{k+1}|\vec{x}_k)f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k})f_{\vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_{k+1}|\vec{y}_{1:N})}{f_{\vec{X}_{k+1}|\vec{Y}_{1:k}}(\vec{x}_{k+1}|\vec{y}_{1:k})} \quad (15.26)$$

Thus, the **smoothing step** recursively updates the **state smoothing posterior PDF**, $f_{\vec{X}_k|\vec{Y}_{1:N}}(\vec{x}_k|\vec{y}_{1:N})$, at time step k by marginalizing out \vec{X}_{k+1} , i.e.

$$f_{\vec{X}_k|\vec{Y}_{1:N}}(\vec{x}_k|\vec{y}_{1:N}) = f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k}) \int \cdots \int \frac{f_{\vec{X}_{k+1}|\vec{X}_k}(\vec{x}_{k+1}|\vec{x}_k)f_{\vec{X}_{k+1}|\vec{Y}_{1:N}}(\vec{x}_{k+1}|\vec{y}_{1:N})}{f_{\vec{X}_{k+1}|\vec{Y}_{1:k}}(\vec{x}_{k+1}|\vec{y}_{1:k})} d\vec{x}_{k+1} \quad (15.27)$$

where $f_{\vec{X}_k|\vec{Y}_{1:k}}(\vec{x}_k|\vec{y}_{1:k})$ is the **state filtering posterior PDF** at time step k and $f_{\vec{X}_{k+1}|\vec{Y}_{1:k}}(\vec{x}_{k+1}|\vec{y}_{1:k})$ is the **state filtering prior PDF** at time step $k + 1$.

Joint State and Parameter Estimation

In the Bayesian estimation framework, one can also incorporate **joint state and parameter estimation** which jointly estimates the unknown *static* model parameters, $\vec{\beta}$, and *dynamic* state, \vec{x} . Then, the joint state and parameter smoothing posterior PDF is

$$f_{\vec{X}_{0:N}, \vec{B} | \vec{Y}_{1:N}}(\vec{x}_{0:N}, \vec{\beta} | \vec{y}_{1:N}) = \frac{f_{\vec{Y}_{1:N} | \vec{X}_{0:N}, \vec{B}}(\vec{y}_{1:N} | \vec{x}_{0:N}, \vec{\beta}) f_{\vec{X}_{0:N} | \vec{B}}(\vec{x}_{0:N} | \vec{\beta}) f_{\vec{B}}(\vec{\beta})}{f_{\vec{Y}_{1:N}}(\vec{y}_{1:N})} \quad (15.28)$$

or

$$f_{\vec{X}_{0:N}, \vec{B} | \vec{Y}_{1:N}}(\vec{x}_{0:N}, \vec{\beta} | \vec{y}_{1:N}) = \frac{\left(\prod_{k=1}^N f_{\vec{Y}_k | \vec{X}_k, \vec{B}}(\vec{y}_k | \vec{x}_k, \vec{\beta}) \right) \left[\left(\prod_{k=1}^N f_{\vec{X}_k | \vec{X}_{k-1}, \vec{B}}(\vec{x}_k | \vec{x}_{k-1}, \vec{\beta}) \right) f_{\vec{X} | \vec{B}}(\vec{x}_0 | \vec{\beta}) \right] f_{\vec{B}}(\vec{\beta})}{f_{\vec{Y}_{1:N}}(\vec{y}_{1:N})} \quad (15.29)$$

One can form the **marginal state posterior PDF**

$$f_{\vec{X}_{0:N} | \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{y}_{1:N}) = \int \cdots \int f_{\vec{X}_{0:N}, \vec{B} | \vec{Y}_{1:N}}(\vec{x}_{0:N}, \vec{\beta} | \vec{y}_{1:N}) d\vec{\beta} \quad (15.30)$$

or

$$f_{\vec{X}_{0:N} | \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{y}_{1:N}) = \int \cdots \int f_{\vec{X}_{0:N} | \vec{B}, \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{\beta}, \vec{y}_{1:N}) f_{\vec{B} | \vec{Y}_{1:N}}(\vec{\beta} | \vec{y}_{1:N}) d\vec{\beta} \quad (15.31)$$

Similarly one can form the **marginal parameter posterior PDF** as

$$f_{\vec{B} | \vec{Y}_{1:N}}(\vec{\beta} | \vec{y}_{1:N}) = \int \cdots \int f_{\vec{X}_{0:N}, \vec{B} | \vec{Y}_{1:N}}(\vec{x}_{0:N}, \vec{\beta} | \vec{y}_{1:N}) d\vec{x}_{0:N} \quad (15.32)$$

or

$$f_{\vec{B} | \vec{Y}_{1:N}}(\vec{\beta} | \vec{y}_{1:N}) = \int \cdots \int f_{\vec{B} | \vec{Y}_{1:N}}(\vec{\beta} | \vec{y}_{1:N}) f_{\vec{X}_{0:N} | \vec{Y}_{1:N}}(\vec{x}_{0:N} | \vec{y}_{1:N}) d\vec{x}_{0:N} \quad (15.33)$$

which can be considered as

$$f_{\vec{B} | \vec{Y}_{1:N}}(\vec{\beta} | \vec{y}_{1:N}) = \frac{f_{\vec{Y}_{1:N} | \vec{B}}(\vec{y}_{1:N} | \vec{\beta}) f_{\vec{B}}(\vec{\beta})}{f_{\vec{Y}_{1:N}}(\vec{y}_{1:N})} \quad (15.34)$$

or by the conditional independence of measurements, one has

$$f_{\vec{B} | \vec{Y}_{1:N}}(\vec{\beta} | \vec{y}_{1:N}) = \frac{\left(\prod_{k=1}^N f_{\vec{Y}_k | \vec{Y}_{1:k-1}, \vec{B}}(\vec{y}_k | \vec{y}_{1:k-1}, \vec{\beta}) \right) f_{\vec{B}}(\vec{\beta})}{f_{\vec{Y}_{1:N}}(\vec{y}_{1:N})} \quad (15.35)$$

where $f_{\vec{Y}_k | \vec{Y}_{1:0}, \vec{B}}(\vec{y}_k | \vec{y}_{1:0}, \vec{\beta}) = f_{\vec{Y}_1 | \vec{B}}(\vec{y}_1 | \vec{\beta})$. This marginal parameter posterior PDF is often easier to estimate using use with Bayes parameter estimation methods than the joint state and parameter estimation using the joint state and parameter posterior PDF.

This problem can also be written as a stochastic state-space system conditioned on the parameter vector, i.e.,

$$\begin{aligned} \vec{x}_k &= f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}; \vec{\beta}) \\ \vec{y}_k &= h_k(\vec{x}_k, \vec{v}_k; \vec{\beta}) \end{aligned} \quad (15.36)$$

For recursive estimation, i.e. filtering, this requires coupling the state and parameter estimation problems into one optimal Bayesian filtering problem and can be approximated in one of two ways using a stochastic state-space framework.

The first is as a **dual estimation** problem which sets up two filters to run in parallel where the state-space system model for the state filter remains the same and the state-space system model for the parameter filter can be setup as

$$\begin{aligned}\vec{\beta}_k &= \vec{\beta}_{k-1} + \vec{e}_{k-1} \\ \vec{y}_k &= h_k(f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}; \vec{\beta}_{k-1}), \vec{v}_k; \vec{\beta}_k)\end{aligned}\quad (15.37)$$

where \vec{e}_{k-1} is some additional process noise in the parameter estimate. Here the posterior state and parameter estimates, $\vec{x}_{k|k}$ and $\vec{\beta}_{k|k}$, are updated in the other parallel filter after each time step.

The second is as a **joint estimation** problem which sets up one filter for a single *joint state* of the state and parameters, i.e. $[\vec{x}^T \quad \vec{\beta}^T]^T$, also known as the **augmented state**, which results in the following **joint state-space model**, also known as the **augmented state-space model**

$$\begin{bmatrix} \vec{x}_k \\ \vec{\beta}_k \end{bmatrix} = \begin{bmatrix} f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}; \vec{\beta}_{k-1}) \\ \vec{\beta}_{k-1} + \vec{e}_{k-1} \end{bmatrix} \quad (15.38)$$

$$\vec{y}_k = h_k(\vec{x}_k, \vec{v}_k; \vec{\beta}_k)$$

In the joint estimation case, even if one has a linear relationship between the output and the parameters and the output and the state, the joint system will be a nonlinear system which requires the use of nonlinear filters, the subject of the next chapter.

Optimal Information Fusion

A **multi-sensor system** that processes multiple sensor output signals to produce information is said to be using **multi-sensor information fusion**, which can be data-driven and/or model-based. This textbook focuses on model-based perception system design using optimal state estimation theory and detection theory. The use of a model-based approach allows one to also robustly assess perception system risk based on model analysis. Model-based multi-sensor information fusion relies heavily on model-based parameter and state estimation methods as this framework allows one to fuse different sensor measurements into an overall state estimate, \hat{x} , where the state is any quantity of interest to the user or the information system.

Therefore, Bayesian multi-sensor data fusion uses a **time update step** from a proprioceptive sensor as a *prediction step* in the traditional Bayes filter context and a **measurement update step** from an exteroceptive sensor as a *correction step* in the traditional Bayes filter context. Furthermore, it should be noted that any number of sensors can be used in this manner by simply employing the appropriate step of the chosen filter. One could also include a known state dynamics model in this data fusion algorithm as an additional *prediction step*. Thus, the Bayes filter equations form the **optimal information fusion** algorithm assuming the sensor models are correct. Furthermore, for truly linear models between the sensor data and the state(s) and AWGN for the sensor error models, the Kalman filter equations provide the **optimal information fusion** algorithm. However, with multi-sensor systems, one can also use a cascade of Bayesian filters and fuse the state estimates of the two or more filters, i.e., **loosely-coupled fusion** at the state estimate level as opposed to **tightly-coupled fusion** at the measurement level. When a steady-state gain is used to fuse scalar information

from two sensors, this is also known as a **complementary filter**, which is typically designed in the frequency domain.

References

For more information, please refer to the following

- Higgins, W. T., “A Comparison of Complementary and Kalman Filtering,” in *IEEE Transactions on Aerospace and Electronic Systems*, Vol. 11, No. 3, 1975, pp. 321-325
- Sarkka, S., “1.3 Optimal filtering and smoothing as Bayesian inference,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 8-12
- Sarkka, S., “1.5 Parameter estimation,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 14-16
- Sarkka, S., “4.1 Probabilistic state space models,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 51-53
- Sarkka, S., “4.2 Bayesian filtering equations,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 54-56
- Sarkka, S., “8.1 Bayesian smoothing equations,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 54-55
- Sarkka, S., “12.1 Bayesian estimation of parameters in state space models,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 174-16
- Simon, D., “15.1 Bayesian state estimation,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 462-466

15.2 Discrete-Time Kalman Filtering and Smoothing

Consider the following multivariate Gaussian PDFs for the state transition PDF and the likelihood PDF

$$\begin{aligned} f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}_k|\vec{x}_{k-1}) &= f_N(\vec{x}_k; F_{k-1}\vec{x}_{k-1} + G_{k-1}\vec{u}_{k-1}, Q_{k-1}) \\ f_{\vec{Y}_k|\vec{X}_k}(\vec{y}_k|\vec{x}_k) &= f_N(\vec{y}_k; H_k \vec{x}_k, R_k) \end{aligned} \quad (15.39)$$

which corresponds to the discrete-time linear stochastic state-space model with additive (weakly) white Gaussian noise (AWGN), also known as the linear-Gaussian state-space model,

$$\begin{aligned} \vec{x}_k &= F_{k-1}\vec{x}_{k-1} + G_{k-1}\vec{u}_{k-1} + \vec{w}_{k-1} \\ \vec{y}_k &= H_k \vec{x}_k + \vec{v}_k \end{aligned} \quad (15.40)$$

where the process noise is $\vec{W}_k \sim \mathcal{N}(0, Q_k)$ and the measurement noise is $\vec{V}_k \sim \mathcal{N}(0, R_k)$ which are uncorrelated with each other.

For linear models with additive white Gaussian noise (AWGN), the state filtering prior and posterior PDFs of the Bayes filter formulation remain multivariate Gaussians for both the prediction and correction steps. This can be proven directly from the Chapman-Kolmogorov equation for two multivariate Gaussians that remains a multivariate Gaussian and for Bayes rule, the multivariate Gaussian is a conjugate prior of the multivariate Gaussian likelihood. However, this section will use two properties for multivariate Gaussians in the Kalman filter derivation which illustrate the relationships between joint multivariate Gaussians, their marginals, and their conditionals.

The first property is if \vec{X} is a multivariate Gaussian distributed as

$$\vec{X} \sim \mathcal{N}(\vec{\mu}_x, \Sigma_x) \quad (15.41)$$

and \vec{Y} is a conditional multivariate Gaussian distributed as

$$\vec{Y}|\vec{X} \sim \mathcal{N}\left(M\vec{x} + \vec{b}, \Sigma_b\right) \quad (15.42)$$

then, one has a joint multivariate Gaussian distribution as

$$\begin{bmatrix} \vec{X} \\ \vec{Y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \vec{\mu}_x \\ M\vec{\mu}_x + \vec{b} \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_x M^T \\ M\Sigma_x & M\Sigma_x M^T + \Sigma_b \end{bmatrix}\right) \quad (15.43)$$

and a marginal multivariate Gaussian distribution as

$$\vec{Y} \sim \mathcal{N}\left(M\vec{\mu}_x + \vec{b}, M\Sigma_x M^T + \Sigma_b\right) \quad (15.44)$$

The second property is if \vec{X} and \vec{Y} are joint multivariate Gaussians distributed as

$$\begin{bmatrix} \vec{X} \\ \vec{Y} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \vec{\mu}_x \\ \vec{\mu}_y \end{bmatrix}, \begin{bmatrix} \Sigma_x & \Sigma_{xy} \\ \Sigma_{xy}^T & \Sigma_y \end{bmatrix}\right) \quad (15.45)$$

then, one has marginal Gaussians distributed as

$$\vec{X} \sim \mathcal{N}(\vec{\mu}_x, \Sigma_x) \quad (15.46)$$

and

$$\vec{Y} \sim \mathcal{N}(\vec{\mu}_y, \Sigma_y) \quad (15.47)$$

and conditional multivariate Gaussians distributed as

$$\vec{X}|\vec{Y} \sim \mathcal{N}\left(\vec{\mu}_x + \Sigma_{xy}\Sigma_y^{-1}(\vec{y} - \vec{\mu}_y), \Sigma_x - \Sigma_{xy}\Sigma_y^{-1}\Sigma_{xy}^T\right) \quad (15.48)$$

and

$$\vec{Y}|\vec{X} \sim \mathcal{N}\left(\vec{\mu}_y + \Sigma_{xy}^T\Sigma_x^{-1}(\vec{x} - \vec{\mu}_x), \Sigma_y - \Sigma_{xy}^T\Sigma_x^{-1}\Sigma_{xy}\right) \quad (15.49)$$

Discrete-Time Kalman Filter

The discrete-time Kalman filter recursion assumes one has knowledge of the state filtering posterior at the previous time step $k - 1$, i.e.,

$$\vec{X}_{k-1} | \vec{Y}_{1:k-1} \sim \mathcal{N}(\vec{\mu}_{k-1|k-1}, P_{k-1|k-1}) \quad (15.50)$$

which for $k = 1$, one has the state prior $\vec{X}_0 \sim \mathcal{N}(\vec{\mu}_0, P_0)$. Note that this section and the following sections on Kalman filtering has denoted these recursions by the subscript “ $i|j$ ” which denotes the hyper-parameter update at step i given measurement updates, i.e., correction steps, up to time step j . For the prediction step derivation, consider the previous posterior state distribution

$$\vec{X}_{k-1} | \vec{Y}_{1:k-1} \sim \mathcal{N}(\vec{\mu}_x, \Sigma_x) \quad (15.51)$$

and the conditional state transition distributed as

$$\vec{X}_k | \vec{X}_{k-1}, \vec{Y}_{1:k-1} \sim \mathcal{N}(F_{k-1}\vec{x}_{k-1} + G_{k-1}\vec{u}_{k-1}, Q_{k-1}) \quad (15.52)$$

Then, the first property noted previously provides the state filtering prior distribution as

$$\vec{X}_k | \vec{Y}_{1:k-1} \sim \mathcal{N}(\vec{\mu}_{k|k-1}, P_{k|k-1}) \quad (15.53)$$

where

$$\vec{\mu}_{k|k-1} = F_{k-1}\vec{\mu}_{k-1} + G_{k-1}\vec{u}_{k-1} \quad (15.54)$$

and

$$P_{k|k-1} = F_{k-1}P_{k-1}F_{k-1}^T + Q_{k-1} \quad (15.55)$$

For the correction step derivation, consider also the measurement likelihood distributed as

$$\vec{Y}_k | \vec{X}_k, \vec{Y}_{1:k-1} \sim \mathcal{N}(H_k\vec{x}_k, R_k) \quad (15.56)$$

Then, the first property noted previously provides the joint distribution of the state and measurement as

$$\vec{X}_k, \vec{Y}_k | \vec{Y}_{1:k-1} = \mathcal{N}\left(\begin{bmatrix} \vec{\mu}_{k|k-1} \\ H_k\vec{\mu}_{k|k-1} \end{bmatrix}, \begin{bmatrix} P_{k|k-1} & P_{k|k-1}H_k^T \\ H_kP_{k|k-1} & H_kP_{k|k-1}H_k^T + R_k \end{bmatrix}\right) \quad (15.57)$$

and the measurement filtering posterior distribution as

$$\vec{Y}_k | \vec{Y}_{1:k-1} \sim \mathcal{N}(H_k\vec{\mu}_{k|k-1}, S_k) \quad (15.58)$$

where

$$S_k = H_kP_{k|k-1}H_k^T + R_k \quad (15.59)$$

Also, the second property noted previously for the joint distribution further provides the state filtering posterior distribution as

$$\vec{X}_k | \vec{Y}_{1:k} \sim \mathcal{N}(\vec{\mu}_{k|k}, P_{k|k}) \quad (15.60)$$

where

$$\vec{\mu}_{k|k} = \vec{\mu}_{k|k-1} + P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1}(\vec{y}_k - H_k\vec{\mu}_{k|k-1}) \quad (15.61)$$

and

$$P_{k|k} = P_{k|k-1} - H_k P_{k|k-1} (H_k P_{k|k-1} H_k^T + R_k)^{-1} P_{k|k-1} H_k^T \quad (15.62)$$

Furthermore, as the mode, mean, and median of a multivariate Gaussian is its mean hyper-parameter, the Kalman filter assumes that the mean recursion is also the state estimate recursion, i.e., $\hat{\vec{x}} = \vec{\mu}$ with the state estimate covariance, P . Thus, from a Bayesian perspective, the discrete-time Kalman filter recursively updates the ‘‘hyper-parameters’’ of the multivariate Gaussian PDF model for the state filtering prior PDF as

$$f_{\vec{X}_k | \vec{Y}_{1:k-1}}(\vec{x}_k | \vec{y}_{1:k-1}) = f_N(\vec{x}_k; \hat{\vec{x}}_{k|k-1}, P_{k|k-1}) \quad (15.63)$$

the measurement filtering posterior PDF as

$$f_{\vec{Y}_k | \vec{Y}_{1:k-1}}(\vec{y}_k | \vec{y}_{1:k-1}) = f_N(\vec{y}_k; H_k \hat{\vec{x}}_{k|k-1}, S_k) \quad (15.64)$$

and the state filtering posterior PDF as

$$f_{\vec{X}_k | \vec{Y}_{1:k}}(\vec{x}_k | \vec{y}_{1:k}) = f_N(\vec{x}_k; \hat{\vec{x}}_{k|k}, P_{k|k}) \quad (15.65)$$

through algebraic recursions for the means and covariances based on the assumed zero-mean noise covariances, Q_k and R_k , the state, input, and measurement matrices, F_k , G_k , and H_k , and the observed inputs and measurements, \vec{u}_k and \vec{y}_k , as functions of the time step, k .

In summary, the discrete-time **KF initialization step** sets the state estimate and state covariance at $k = 0$ as

$$\begin{aligned} \hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 \\ P_{0|0} &= P_0 \end{aligned} \quad (15.66)$$

The discrete-time **KF prediction step** predicts the state estimate, the state covariance, and the output using the input and state-space model parameters as

$$\begin{aligned} \hat{\vec{x}}_{k|k-1} &= F_{k-1} \hat{\vec{x}}_{k-1|k-1} + G_{k-1} \vec{u}_{k-1} \\ P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \\ \hat{\vec{y}}_k &= H_k \hat{\vec{x}}_{k|k-1} \end{aligned} \quad (15.67)$$

The discrete-time **KF correction step** updates the state estimator and the state covariance using the measurement and state-space model parameters as

$$\begin{aligned} \tilde{y}_k &= \vec{y}_k - \hat{\vec{y}}_k \\ S_k &= H_k P_{k|k-1} H_k^T + R_k \\ K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k \tilde{y}_k \\ P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T \end{aligned} \quad (15.68)$$

where \tilde{y}_k is the zero-mean **innovation process**, also known as the ‘‘pre-fit residual,’’ with S_k as its **innovation covariance**. By inspection, the KF correction step is the same as recursive least-squares parameter estimator

while the additional prediction step provides the least-squares predictor as well. Furthermore, by extension of least-squares analysis, the Kalman filter is the MMSE, the MVUE, and the BLUE for the linear state-space model without the explicit Gaussian noise assumption. One simply only requires knowledge of exact zero-mean mean and covariances for the noise statistics and model matrices.

Furthermore, note that by definition of the posterior state covariance, one has

$$P_{k|k} = \text{Cov} \left(\vec{x}_k - \hat{\vec{x}}_{k|k} \right) \quad (15.69)$$

and by substitution for $\hat{\vec{x}}_{k|k} = \hat{\vec{x}}_{k|k-1} + K_k \tilde{y}_k$, $\tilde{y}_k = \vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}$, and $\vec{y}_k = H_k \vec{x}_k + \vec{v}_k$, one has

$$P_{k|k} = \text{Cov} \left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} + K_k (H_k \vec{x}_k + \vec{v}_k - H_k \hat{\vec{x}}_{k|k-1}) \right) \quad (15.70)$$

$$P_{k|k} = \text{Cov} \left((I - K_k H_k) (\vec{x}_k - \hat{\vec{x}}_{k|k-1}) - K_k \vec{v}_k \right) \quad (15.71)$$

$$P_{k|k} = (I - K_k H_k) \text{Cov} \left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} \right) (I - K_k H_k)^T + K_k \text{Cov}(\vec{v}_k) K_k^T \quad (15.72)$$

Then, by definition, one has the **Joseph form** of the state estimate covariance posterior correction recursion

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T \quad (15.73)$$

which holds for any gain K_k .

Notably, one can expand the Joseph form out as

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} - P_{k|k-1} H_k^T K_k^T + K_k H_k P_{k|k-1} H_k^T K_k^T + K_k R_k K_k^T \quad (15.74)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} - P_{k|k-1} H_k^T K_k^T + K_k (H_k P_{k|k-1} H_k^T + R_k) K_k^T \quad (15.75)$$

By substitution of the variance-minimizing k_k and $S_k = S_k^T$, one has

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} - P_{k|k-1} H_k^T S_k^{-1} H_k P_{k|k-1} + P_{k|k-1} H_k^T S_k^{-1} H_k P_{k|k-1} \quad (15.76)$$

one obtains another alternative state estimate covariance update equation as

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} \quad (15.77)$$

Furthermore, note that combining these equations provides the **one-step prior discrete-time Kalman filter** as

$$\begin{aligned} \hat{\vec{y}}_k &= H_k \hat{\vec{x}}_{k|k-1} \\ K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \\ \hat{\vec{x}}_{k+1|k} &= F_k \hat{\vec{x}}_{k|k-1} + G_k \vec{u}_k + F_k K_k \left(\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1} \right) \\ P_{k+1|k} &= F_k (I - K_k H_k) P_{k|k-1} F_k^T + Q_k \end{aligned} \quad (15.78)$$

and the **one-step posterior discrete-time Kalman filter** as the Luenberger observer equations for the posterior mean recursion, the Riccati difference equation for the posterior covariance recursion

$$\begin{aligned}\hat{\vec{y}}_k &= H_k \left(F_{k-1} \hat{\vec{x}}_{k-1|k-1} + G_{k-1} \vec{u}_{k-1} \right) \\ K_k &= (F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}) H_k^T \left(H_k (F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1}) H_k^T + R_k \right)^{-1} \\ \hat{x}_{k|k} &= F_{k-1} \hat{\vec{x}}_{k-1|k-1} + G_{k-1} \vec{u}_{k-1} + K_k \left(\vec{y}_k - \hat{\vec{y}}_k \right) \\ P_{k|k} &= (I - K_k H_k) \left(F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \right)\end{aligned}\tag{15.79}$$

One can consider the Kalman gain K_k as the “transformed” relative weighting between previous P , Q , and R to balance the uncertainty one has in the process model versus the measurement model when optimally estimating the state. To see this consider the simplified univariate example with

$$P_{k-1|k-1} = \sigma_x^2, \quad F = 1, \quad H = 1, \quad R = \sigma_R^2, \quad Q = \sigma_Q^2\tag{15.80}$$

which provides

$$P_{k|k-1} = \sigma_x^2 + \sigma_Q^2\tag{15.81}$$

$$S_k = \sigma_x^2 + \sigma_Q^2 + \sigma_R^2\tag{15.82}$$

and

$$K_k = \frac{\sigma_x^2 + \sigma_Q^2}{\sigma_x^2 + \sigma_Q^2 + \sigma_R^2}\tag{15.83}$$

which is large when the covariance in the process is relatively larger than the covariance of the measurement, i.e. $\sigma_R^2 \ll \sigma_Q^2$.

Discrete-Time Fixed-Interval Kalman Smoothing

The **fixed-interval Kalman smoother (FI-KS)**, also known as the **Rauch-Tung-Striebel smoother (RTSS)**, assumes one has knowledge of the state filtering prior distribution as

$$\vec{X}_{k+1} | \vec{Y}_{1:k} \sim \mathcal{N} \left(\hat{\vec{x}}_{k+1|k}, P_{k+1|k} \right)\tag{15.84}$$

the state filtering posterior distribution as

$$\vec{X}_k | \vec{Y}_{1:k} \sim \mathcal{N} \left(\hat{\vec{x}}_{k|k}, P_{k|k} \right)\tag{15.85}$$

and the state smoothing posterior distribution as

$$\vec{X}_k | \vec{Y}_{1:N} \sim \mathcal{N} \left(\hat{\vec{x}}_{k+1|N}, P_{k+1|N} \right)\tag{15.86}$$

which for $k+1 = N$, one has the initial state smoothing posterior equivalent to the final state filtering posterior, i.e., $\vec{X}_N | \vec{Y}_{1:N} \sim \mathcal{N}(\hat{\vec{x}}_{N|N}, P_{N|N})$.

To this end, consider the previous posterior state distribution

$$\vec{X}_k | \vec{Y}_{1:k} \sim \mathcal{N} \left(\hat{\vec{x}}_{k|k}, P_{k|k} \right) \quad (15.87)$$

and the conditional distribution

$$\vec{X}_{k+1} | \vec{X}_k, \vec{Y}_{1:k} \sim \mathcal{N} \left(F_k \hat{\vec{x}}_k + G_k \vec{u}_k, Q_k \right) \quad (15.88)$$

Then, the first property noted previously provides the joint distribution of consecutive states as

$$\begin{bmatrix} \vec{X}_k | \vec{Y}_{1:k} \\ \vec{X}_{k+1} | \vec{X}_k, \vec{Y}_{1:k} \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \hat{\vec{x}}_{k|k} \\ \hat{\vec{x}}_{k+1|k} \end{bmatrix}, \begin{bmatrix} P_{k|k} & P_{k|k} F_k^T \\ F_k P_{k|k} & P_{k|k-1} \end{bmatrix} \right) \quad (15.89)$$

By the Markov assumption, the second element can be rewritten over the entire measurement history as

$$\begin{bmatrix} \vec{X}_k | \vec{Y}_{1:k} \\ \vec{X}_{k+1} | \vec{X}_k, \vec{Y}_{1:N} \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \hat{\vec{x}}_{k|k} \\ \hat{\vec{x}}_{k+1|k} \end{bmatrix}, \begin{bmatrix} P_{k|k} & P_{k|k} F_k^T \\ F_k P_{k|k} & P_{k|k-1} \end{bmatrix} \right) \quad (15.90)$$

which by the second property noted previously for the joint distribution further provides the marginal distribution as

$$\vec{X}_k | \vec{X}_{k+1}, \vec{Y}_{1:N} \sim \mathcal{N} \left(\hat{\vec{x}}_{k|k} + K_{S,k} (\vec{x}_{k+1} - \hat{\vec{x}}_{k|k}), P_{k|k} - K_{S,k} P_{k|k-1} K_{S,k}^T \right) \quad (15.91)$$

with

$$K_{S,k} = P_{k|k} F_k^T P_{k|k-1}^{-1} \quad (15.92)$$

Then, for the joint smoothed distribution for consecutive states, one has

$$\begin{bmatrix} \vec{X}_{k+1} | \vec{Y}_{1:N} \\ \vec{X}_k | \vec{Y}_{1:N} \end{bmatrix} \sim \mathcal{N} \left(\begin{bmatrix} \hat{\vec{x}}_{k+1|N} \\ \hat{\vec{x}}_{k|k} + K_{S,k} (\hat{\vec{x}}_{k+1|N} - \hat{\vec{x}}_{k+1|k}) \end{bmatrix}, \begin{bmatrix} P_{k+1|N} & P_{k+1|N} K_{S,k}^T \\ K_{S,k} P_{k+1|N} & K_{S,k} P_{k+1|N} K_{S,k}^T + P_{k|k} - K_{S,k} P_{k|k-1} K_{S,k}^T \end{bmatrix} \right) \quad (15.93)$$

and by the second property noted previously provides the state smoothing posterior distributed as

$$\vec{X}_k | \vec{Y}_{1:N} \sim \mathcal{N} \left(\hat{\vec{x}}_{k|k} + K_{S,k} (\hat{\vec{x}}_{k+1|N} - \hat{\vec{x}}_{k+1|k}), P_{k|k} + K_{S,k} (P_{k+1|N} - P_{k|k-1}) K_{S,k}^T \right) \quad (15.94)$$

where one can alternatively keep in memory the state filtering posterior mean and covariance and the input, i.e.,

$$\vec{X}_k | \vec{Y}_{1:N} \sim \mathcal{N} \left(\hat{\vec{x}}_{k|k} + K_{S,k} (\hat{\vec{x}}_{k+1|N} - F_k \hat{\vec{x}}_{k|k} - G_k \vec{u}_k), P_{k|k} + K_{S,k} (P_{k+1|N} - F_k P_{k|k} F_k^T - Q_k) K_{S,k}^T \right) \quad (15.95)$$

Thus, the discrete-time **FI-KS smoothing step** or the **RTSS smoothing step** updates the state estimate and the state covariance using the posterior state estimate and covariance and the prior state estimate and

covariance or the measurement noise covariance as

$$\begin{aligned}
 \tilde{\vec{x}}_k &= \hat{\vec{x}}_{k+1|N} - \hat{\vec{x}}_{k+1|k} \\
 &= \hat{\vec{x}}_{k+1|N} - F_k \hat{\vec{x}}_{k|k} - G_k \vec{u}_k \\
 K_{S,k} &= P_{k|k} F_k^T P_{k+1|k}^{-1} \\
 &= P_{k|k} F_k^T (F_k P_{k|k} F_k^T + Q_k)^{-1} \\
 \hat{\vec{x}}_{k|N} &= \hat{\vec{x}}_{k|k} + K_{S,k} \tilde{\vec{x}}_k \\
 P_{k|N} &= P_{k|k} + K_{S,k} (P_{k+1|N} - P_{k+1|k}) K_{S,k}^T \\
 &= P_{k|k} + K_{S,k} (P_{k+1|N} - F_k P_{k|k} F_k^T - Q_k) K_{S,k}^T
 \end{aligned} \tag{15.96}$$

References

For more information, please refer to the following

- Sarkka, S., “4.3 Kalman Filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 56-62
- Sarkka, S., “8.2 Rauch-Tung-Striebel Smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 135-139
- Simon, D., “5.1 Derivation of the Discrete-Time Kalman Filter,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 124-129
- Simon, D., “9.4.2 RTS Smoothing,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 286-294

15.3 Alternative Discrete-Time Kalman Filtering and Smoothing

The standard Kalman filter equations assume a purely mathematical derivation from the principles of probability theory. However in many cases, the use of the Kalman filter in embedded systems may require special consideration with respect to computational resources and/or numerical precision. However, due to the linearity of the Kalman filter, one can derive a variety of alternative recursions for the prediction and correction steps of the Kalman depending on additional considerations. In addition, the memory requirements for a large number of measurements or time steps may not be feasible, especially for fixed-interval Kalman smoothing requiring memory for each state filtering prior and posterior state estimate and covariance. Another drawback of fixed-interval smoothing is the recalculation required if additional measurements are obtained and N increases, e.g., for M additional measurements, M prediction, correction, and smoothing steps must be computed and N smoothing steps must be recomputed. Thus, an alternative for smoothing in real-time applications is a fixed-lag Kalman smoother which augments the Kalman filter state and smooths the state estimate at a fixed-lag of $k - N$ time steps behind the current measurement at k .

For this section, recall the standard discrete-time Kalman filter algorithm can be described as an initialization step

$$\begin{aligned}\hat{x}_{0|0} &= \hat{x}_0 \\ P_{0|0} &= P_0\end{aligned}\tag{15.97}$$

a prediction step

$$\begin{aligned}\hat{x}_{k|k-1} &= F_{k-1} \hat{x}_{k-1|k-1} + G_{k-1} \vec{u}_{k-1} \\ P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \\ \hat{y}_k &= H_k \hat{x}_{k|k-1}\end{aligned}\tag{15.98}$$

and a correction step

$$\begin{aligned}S_k &= H_k P_{k|k-1} H_k^T + R_k \\ K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k (\vec{y}_k - H_k \hat{x}_{k|k-1}) \\ P_{k|k} &= (I - K_k H_k) P_{k|k-1}\end{aligned}\tag{15.99}$$

where notably the inversion of S_k^{-1} , an $n_y \times n_y$ matrix, is required which has computational complexity $\mathcal{O}(n_y^3)$ using the common Gauss-Jordan elimination method or slightly lower for other methods. Thus, for large n_y , one may prefer an alternative to the inversion and memory requirements for the S_k and R_k matrix mathematics operations in the Kalman filter.

Sequential Kalman Filtering

One alternative to reducing the inversion computational complexity and the memory requirements for large n_y is to sequentially correct for subsets of measurements which is possible if the measurement noise covariance is block diagonal with r blocks, i.e.,

$$R_k = \begin{bmatrix} R_{k,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{k,r} \end{bmatrix}\tag{15.100}$$

which invites the following partitioning of the measurement equation of the linear-Gaussian state-space model as

$$\begin{bmatrix} \vec{y}_{k,1} \\ \vdots \\ \vec{y}_{k,r} \end{bmatrix} = \begin{bmatrix} H_{k,1} \\ \vdots \\ H_{k,r} \end{bmatrix} \vec{x}_k + \begin{bmatrix} \vec{v}_{k,1} \\ \vdots \\ \vec{v}_{k,r} \end{bmatrix}\tag{15.101}$$

with $\vec{v}_{k,i} \sim \mathcal{N}(0, R_{k,i})$ for $i = 1, \dots, r$.

Thus, the **discrete-time sequential Kalman filter (SKF)** algorithm has a modified correction step which is initialized with the prior state estimate and covariance

$$\begin{aligned}\hat{x}_{k|k-1,0} &= \hat{x}_{k|k-1} \\ P_{0,k|k-1} &= P_{k|k-1}\end{aligned}\tag{15.102}$$

and runs $i = 1, \dots, r$ iterations of

$$\begin{aligned} S_{k,i} &= H_{k,i} P_{k|k-1,i-1} H_{k,i}^T + R_{k,i} \\ K_{k,i} &= P_{k|k-1,i-1} H_{k,i}^T S_{k,i}^{-1} \\ \hat{x}_{k|k-1,i} &= \hat{x}_{k|k-1,i} + K_{k,i} (\vec{y}_{k,i} - H_{k,i} \hat{x}_{k|k-1,i}) \\ P_{k|k-1,i} &= (I - K_{k,i} H_{k,i}) P_{k|k-1,i-1} \end{aligned} \quad (15.103)$$

and assigns for the posterior state estimate and covariance

$$\begin{aligned} \hat{x}_{k|k} &= \hat{x}_{k|k-1,r} \\ P_{0,k|k-1} &= P_{k|k-1,r} \end{aligned} \quad (15.104)$$

Note that if R is diagonal, then no inversions are necessary. Furthermore, if $R_k = R$, one can obtain a Jordan form decomposition for $\tilde{R} = V^T RV$ with \tilde{R} containing block matrices which provides the transformed measurement model

$$\tilde{\vec{y}} = \tilde{H}_k \vec{x}_k + \tilde{\vec{v}} \quad (15.105)$$

with $\tilde{H}_k = V^T H_k$ and $\tilde{\vec{v}} \sim \mathcal{N}(0, \tilde{R})$ which can be used in the SKF.

Information Filtering

Another alternative recursion for the Kalman filter to change the inversion computation is to propagate the inverse of the state covariance matrix, P , i.e., the **information matrix**, $\mathcal{I} = P^{-1}$ which forms the **information filter (IF)**. then, by substitution, the discrete-time correction step update for the information matrix is

$$\mathcal{I}_{k|k} = \mathcal{I}_{k|k-1} + H_k^T R_k^{-1} H_k \quad (15.106)$$

and the discrete-time prediction step update for the information matrix is

$$\mathcal{I}_{k|k-1} = \left(F_{k-1} \mathcal{I}_{k-1|k-1}^{-1} F_k^T + Q_{k-1} \right)^{-1} \quad (15.107)$$

Next, applying the matrix inversion lemma, i.e.,

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(DA^{-1}B + C^{-1})^{-1}DA^{-1} \quad (15.108)$$

where $A = Q_{k-1}$, $B = F_{k-1}$, $C = \mathcal{I}^{-1}$, and $D = F_{k-1}^T$, one has

$$\mathcal{I}_{k|k-1} = Q_{k-1}^{-1} - Q_{k-1}^{-1} F_{k-1} \left(\mathcal{I}_{k-1|k-1} + F_{k-1}^T Q_{k-1}^{-1} F_{k-1} \right) F_{k-1}^T Q_{k-1}^{-1} \quad (15.109)$$

The information filter equations requires a couple of $n_x \times n_x$ inversions. Therefore, if $n_y \gg n_x$, it may be more efficient to use the information filter. Also, if R_k or Q_{k-1} are constant with respect to k , then one could invert those as part of the initialization process. Furthermore, if the initial uncertainty is zero, one can numerically set $\mathcal{I}_0 = \infty$. If the initial uncertainty is ∞ , one can numerically set $P_0 = \infty$.

In summary, the **IF prediction step** can be written as

$$\begin{aligned}\hat{x}_{k|k-1} &= F_{k-1}\hat{x}_{k-1} + G_k \vec{u}_{k-1} \\ \mathcal{I}_{k|k-1} &= Q_{k-1}^{-1} - Q_{k-1}^{-1}F_{k-1} \left(\mathcal{I}_{k-1|k-1} + F_{k-1}^T Q_{k-1}^{-1} F_{k-1} \right) F_{k-1}^T Q_{k-1}^{-1} \\ \hat{\vec{y}}_k &= H_k \hat{x}_{k|k-1}\end{aligned}\quad (15.110)$$

and the **IF correction step** can be written as

$$\begin{aligned}\tilde{y}_k &= \vec{y}_k - \hat{\vec{y}}_k \\ \mathcal{I}_{k|k} &= \mathcal{I}_{k|k-1} + H_k^T R_k^{-1} H_k \\ K_k &= \mathcal{I}_{k|k}^{-1} H_k^T R_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k\end{aligned}\quad (15.111)$$

Square-Root Kalman Filtering

In Kalman filtering, it can be advantageous to propagate the square-root of the covariance matrix, $P_k^{1/2}$, as opposed to the covariance matrix, P_k . This advantage comes from the increase in numerical precision for the inversions required in the Kalman filter equations, but at the cost of additional computational resources.

To derive the prediction step for the **square-root Kalman filter (SR-KF)**, suppose one can find a $2n_x \times 2n_x$ matrix

$$T = \begin{bmatrix} T_1 & T_2 \end{bmatrix} \quad (15.112)$$

such that one could write the **SR-KF prediction step** for $P_{k|k-1}^{T/2}$ as

$$\begin{bmatrix} P_{k|k-1}^{T/2} \\ 0 \end{bmatrix} = T \begin{bmatrix} P_{k-1|k-1}^{T/2} F^T \\ Q^{T/2} \end{bmatrix} \quad (15.113)$$

then,

$$\begin{bmatrix} P_{k|k-1}^{T/2} \\ 0 \end{bmatrix} = T_1 P_{k-1|k-1}^{T/2} F^T + T_2 Q^{T/2} \quad (15.114)$$

Furthermore, if T is prescribed as orthogonal, one has

$$T^T T = \begin{bmatrix} T_1^T \\ T_2^T \end{bmatrix} \begin{bmatrix} T_1 & T_2 \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (15.115)$$

which allows one to write

$$\begin{bmatrix} P_{k|k-1}^{1/2} & 0 \end{bmatrix} \begin{bmatrix} P_{k|k-1}^{1/2} \\ 0 \end{bmatrix} = \left[T_1 P_{k-1|k-1}^{T/2} F^T + T_2 Q^{T/2} \right]^T \left[T_1 P_{k-1|k-1}^{T/2} F^T + T_2 Q^{T/2} \right] \quad (15.116)$$

which allows one to write

$$P_{k|k-1}^{1/2} P_{k|k-1}^{T/2} = F P_{k-1|k-1}^{1/2} T_1^T T_1 P_{k-1|k-1}^{T/2} F^T + Q^{1/2} T_2^T T_2 Q^{T/2} \quad (15.117)$$

which implies by the orthogonal properties and square-root properties that

$$P_{k|k-1} = FP_{k-1|k-1}F^T + Q \quad (15.118)$$

which is the prediction step update of the covariance and proves the square-root prediction step is equivalent. However, one requires methods for finding T and $P_{k|k-1}^{T/2}$ matrices that solve the square-root prediction step use various methods from linear algebra, e.g. Householder, Gram-Schmidt, modified Gram-Schmidt, or Givens transformations.

For the correction step of the square-root filter, suppose one can find a $(n_x + n_y) \times (n_x + n_y)$ matrix,

$$\tilde{T} = \begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} \\ \tilde{T}_{21} & \tilde{T}_{22} \end{bmatrix} \quad (15.119)$$

such that one can write the **SR-KF correction step** for $P_{k|k-1}^T$ as

$$\begin{bmatrix} (R + HP_{k|k-1}H^T)^{T/2} & \tilde{K}_k \\ 0 & P_{k|k}^{T/2} \end{bmatrix} = \tilde{T} \begin{bmatrix} R^{T/2} & 0 \\ P_{k|k-1}^T H^T & P_{k|k-1}^T \end{bmatrix} \quad (15.120)$$

where

$$\tilde{K}_k = K_k(R + HP_{k|k-1}H^T)^{T/2} \quad (15.121)$$

and the lower right matrix of the block matrix on the left side, i.e. $P_{k|k}^{T/2}$, cannot be found until one can solve for \tilde{T} that provides the other 3 matrices in the block matrix on the left side. Furthermore, if \tilde{T} is prescribed as orthogonal, one has

$$\tilde{T}^T \tilde{T} = \begin{bmatrix} \tilde{T}_{11}^T & \tilde{T}_{21}^T \\ \tilde{T}_{12}^T & \tilde{T}_{22}^T \end{bmatrix} \begin{bmatrix} \tilde{T}_{11} & \tilde{T}_{12} \\ \tilde{T}_{21} & \tilde{T}_{22} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & I \end{bmatrix} \quad (15.122)$$

which allows one to write

$$\begin{bmatrix} (R + HP_{k|k-1}H^T)^{T/2} & \tilde{K}_k \\ 0 & P_{k|k}^{T/2} \end{bmatrix} = \begin{bmatrix} \tilde{T}_{11}R^{T/2} + \tilde{T}_{12}P_{k|k-1}^{T/2}H^T & \tilde{T}_{12}P_{k|k-1}^T \\ \tilde{T}_{21}R^{T/2} + \tilde{T}_{22}P_{k|k-1}^{T/2}H^T & \tilde{T}_{22}P_{k|k-1}^T \end{bmatrix} \quad (15.123)$$

which can be written as four separate equalities by pre-multiplying by its transpose. The first column provides two equations

$$\begin{aligned} (R + HP_{k|k-1}H^T)^{1/2}(R + HP_{k|k-1}H^T)^{T/2} = \\ R^{1/2}\tilde{T}_{11}^T\tilde{T}_{11}R^{T/2} + HP_{k|k-1}^{1/2}\tilde{T}_{12}\tilde{T}_{11}R^{T/2} \\ + R^{1/2}\tilde{T}_{11}^T\tilde{T}_{12}P_{k|k-1}^{T/2}H^T + HP_{k|k-1}^{1/2}\tilde{T}_{12}^T\tilde{T}_{12}P_{k|k-1}^{T/2}H^T \end{aligned} \quad (15.124)$$

and

$$\begin{aligned} 0 = R^{1/2}\tilde{T}_{21}^T\tilde{T}_{21}R^{T/2} + R^{1/2}\tilde{T}_{21}^T\tilde{T}_{22}P_{k|k-1}^T H^T \\ + HP_{k|k-1}^{1/2}\tilde{T}_{22}^T\tilde{T}_{21}R^{T/2} + HP_{k|k-1}^{1/2}\tilde{T}_{22}^T\tilde{T}_{22}P_{k|k-1}^T H^T \end{aligned} \quad (15.125)$$

which adding together and simplifying with the transpose properties provides

$$R + HP_{k|k-1}H^T = R + HP_{k|k-1}^{1/2}P_{k|k-1}^{T/2}H^T \quad (15.126)$$

which shows the square-root property for $P_{k|k-1}^{1/2}$. From the second column, one has

$$\tilde{K}_k \tilde{K}_k^T = P_{k|k-1}^{1/2} \tilde{T}_{12}^T \tilde{T}_{12} P_{k|k-1}^{T/2} \quad (15.127)$$

and

$$P_{k|k}^{1/2} P_{k|k}^{T/2} = P_{k|k-1}^{1/2} \tilde{T}_{22}^T \tilde{T}_{22} P_{k|k-1}^{T/2} \quad (15.128)$$

which adding together and simplifying with the transpose properties provides

$$P_{k|k} + K_k (R + H P_{k|k-1} H^T) K_k^T = P_{k|k-1}^{1/2} P_{k|k-1}^{T/2} \quad (15.129)$$

and substituting for the Kalman gain K_k from the standard equation provides

$$P_{k|k} + P_{k|k-1} H^T K_k^T = P_{k|k-1} \quad (15.130)$$

or rearranging and using the fact that each term is symmetric, one has

$$P_{k|k} == P_{k|k-1} - K_k H P_{k|k-1} \quad (15.131)$$

which is the correction step update of the covariance and proves the square-root correction step is equivalent. However, one requires methods for finding \tilde{T} and $P_{k|k}^{1/2}$ matrices that solve the square-root correction step.

Discrete-Time Fixed-Lag Kalman Smoothing

The **fixed-lag Kalman smoother (FL-KS) step** recursion augments the state-space model by $N + 1$ stacked state vectors to obtain the model

$$\begin{aligned} \begin{bmatrix} \vec{x}_{k+1} \\ \vec{x}_k \\ \vdots \\ \vec{x}_{k-N} \end{bmatrix} &= \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} \vec{x}_k \\ \vec{x}_{k-1} \\ \vdots \\ \vec{x}_{k-N-1} \end{bmatrix} + \begin{bmatrix} G_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \vec{u}_k + \begin{bmatrix} I \\ 0 \\ \vdots \\ 0 \end{bmatrix} \vec{w}_k \\ \vec{y}_k &= [H_k \ 0 \ \cdots \ 0] \begin{bmatrix} \vec{x}_k \\ \vec{x}_k \\ \vdots \\ \vec{x}_{k-N} \end{bmatrix} + \vec{v}_k \end{aligned} \quad (15.132)$$

Next, consider the one-step prior discrete-time Kalman filter recursion as

$$\begin{aligned} K_k &= P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \\ \hat{x}_{k+1|k} &= F_k \hat{x}_{k|k-1} + G_k \vec{u}_k + F_k K_k (\vec{y}_k - H_k \hat{x}_{k|k-1}) \\ P_{k+1|k} &= F_k P_{k|k-1} (I - H_k^T K_k^T) F_k^T + Q_k \end{aligned} \quad (15.133)$$

Then, the one-step prior Kalman filter state estimate for the augmented state-space model can be written as

$$\begin{aligned} \begin{bmatrix} \hat{\vec{x}}_{k+1|k} \\ \hat{\vec{x}}_{k|k} \\ \vdots \\ \hat{\vec{x}}_{k-N|k} \end{bmatrix} &= \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} \hat{\vec{x}}_{k|k-1} \\ \hat{\vec{x}}_{k-1|k-1} \\ \vdots \\ \hat{\vec{x}}_{k-N-1|k-1} \end{bmatrix} + \begin{bmatrix} G_k \\ 0 \\ \vdots \\ 0 \end{bmatrix} \vec{u}_k \\ &+ \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} K_{k,0} \\ K_{k,1} \\ \vdots \\ K_{k,N+1} \end{bmatrix} \left(\vec{y}_k - [H_k \ 0 \ \cdots \ 0] \begin{bmatrix} \hat{\vec{x}}_{k|k-1} \\ \hat{\vec{x}}_{k-1|k-1} \\ \vdots \\ \hat{\vec{x}}_{k-N-1|k-1} \end{bmatrix} \right) \end{aligned} \quad (15.134)$$

or

$$\begin{bmatrix} \hat{\vec{x}}_{k+1|k} \\ \hat{\vec{x}}_{k|k} \\ \hat{\vec{x}}_{k-1|k} \\ \vdots \\ \hat{\vec{x}}_{k-N|k} \end{bmatrix} = \begin{bmatrix} F_k \hat{\vec{x}}_{k|k-1} + G_k \vec{u}_k + F_k K_{k,0} (\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}) \\ \hat{\vec{x}}_{k|k-1} + K_{k,0} (\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}) \\ \hat{\vec{x}}_{k-1|k-1} + K_{k,1} (\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}) \\ \vdots \\ \hat{\vec{x}}_{k-N|k-1} + K_{k,N} (\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}) \end{bmatrix} \quad (15.135)$$

and notably the first entry for this construction provides the one-step Kalman filter state estimate equation the second entry for this construction provides the Kalman filter posterior state estimate correction, $\hat{\vec{x}}_{k|k} = \hat{\vec{x}}_{k|k-1} + K_{k,0}(\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1})$ with $K_{k,0} = K_k$, which is also used to update previous posterior states to step $k - N$ given measurements up to time step k . The additional Kalman gains adjust the previous state estimates with the new innovation $\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}$.

Next, the one-step prior Kalman filter stacked Kalman gains for the augmented state-space model

$$\begin{aligned} \begin{bmatrix} F_k K_{k,0} \\ K_{k,0} \\ \vdots \\ K_{k,N} \end{bmatrix} &= \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} P_{k|k-1} & P_{k,k-1|k-1}^T & \cdots & P_{k,k-N-1|k-1}^T \\ P_{k,k-1|k-1} & P_{k-1|k-1} & \cdots & P_{k,k-N-1|k-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ P_{k,k-N-1|k-1} & P_{k,k-N-1|k-1} & \cdots & P_{k-N-1|k-1} \end{bmatrix} \begin{bmatrix} H_k^T \\ 0 \\ \vdots \\ 0 \end{bmatrix} \\ &\quad \left(\begin{bmatrix} H_k & 0 & \cdots & 0 \end{bmatrix} \begin{bmatrix} P_{k|k-1} & P_{k,k-1|k-1}^T & \cdots & P_{k,k-N-1|k-1}^T \\ P_{k,k-1|k-1} & P_{k-1|k-1} & \cdots & P_{k,k-N-1|k-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ P_{k+1,k-N-1|k-1} & P_{k,k-N-1|k-1} & \cdots & P_{k-N-1|k-1} \end{bmatrix} \begin{bmatrix} H_k^T \\ 0 \\ \vdots \\ 0 \end{bmatrix} + R_k \right)^{-1} \end{aligned} \quad (15.136)$$

which simplifies to

$$\begin{bmatrix} F_k K_{k,0} \\ K_{k,0} \\ K_{k,1} \\ \vdots \\ K_{k,N} \end{bmatrix} = \begin{bmatrix} F_k P_{k|k-1} \\ P_{k|k-1} \\ P_{k,k-1|k-1} \\ \vdots \\ P_{k,k-N|k-1} \end{bmatrix} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1} \quad (15.137)$$

where notably, the second entry, $K_{k,0} = P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1}$ is the standard Kalman gain for the correction step at time step k . The additional Kalman gains use the cross-covariance between k and $i = 1, \dots, N+1$ previous steps.

Then, one-step prior Kalman filter state covariance for the augmented state-space model can be written as

$$\begin{bmatrix} P_{k+1|k} & P_{k+1,k|k}^T & \cdots & P_{k+1,k-N|k}^T \\ P_{k+1,k|k} & P_{k|k} & \cdots & P_{k,k-N|k}^T \\ \vdots & \vdots & \ddots & \vdots \\ P_{k+1,k-N|k} & P_{k,k-N|k} & \cdots & P_{k-N|k} \end{bmatrix} = \begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} \begin{bmatrix} P_{k|k-1} & P_{k,k-1|k-1}^T & \cdots & P_{k,k-N-1|k-1}^T \\ P_{k,k-1|k-1} & P_{k-1|k-1} & \cdots & P_{k-1,k-N-1|k-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ P_{k,k-N-1|k-1} & P_{k,k-N-1|k-1} & \cdots & P_{k-N-1|k-1} \end{bmatrix} \\ \left(\begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} - \begin{bmatrix} F_k K_{k,0} \\ K_{k,0} \\ \vdots \\ K_{k,N+1} \end{bmatrix} \begin{bmatrix} H_k & 0 & \cdots & 0 \end{bmatrix} \right)^T \begin{bmatrix} Q_k & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (15.138)$$

or

$$\begin{bmatrix} P_{k+1|k} & P_{k+1,k|k}^T & \cdots & P_{k+1,k-N|k}^T \\ P_{k+1,k|k} & P_{k|k} & \cdots & P_{k,k-N-1|k}^T \\ \vdots & \vdots & \ddots & \vdots \\ P_{k+1,k-N|k} & P_{k,k-N|k} & \cdots & P_{k-N|k} \end{bmatrix} = \left(\begin{bmatrix} F_k & 0 & \cdots & 0 \\ I & 0 & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & I & 0 \end{bmatrix} - \begin{bmatrix} F_k K_{k,0} H_k & 0 & \cdots & 0 \\ K_{k,0} H_k & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ K_{k,N+1} H_k & 0 & \cdots & 0 \end{bmatrix} \right) \\ \begin{bmatrix} P_{k|k-1} F_k^T & P_{k|k-1} & \cdots & P_{k,k-N|k-1}^T \\ P_{k,k-1|k-1} F_k^T & P_{k,k-1|k-1} & \cdots & P_{k-1,k-N|k-1}^T \\ \vdots & \vdots & \ddots & \vdots \\ P_{k,k-N-1|k-1} F_k^T & P_{k,k-N-1|k-1} & \cdots & P_{k-N,k-N-1|k-1} \end{bmatrix} \\ + \begin{bmatrix} Q_k & 0 & \cdots & 0 \\ 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 \end{bmatrix} \quad (15.139)$$

By inspection, this provides the following recursions for the first column of the covariance matrix are

$$\begin{aligned} P_{k+1|k} &= F_k P_{k|k-1} (I - H_k^T K_{k,0}) F_k^T + Q_k \\ P_{k+1,k|k} &= P_{k|k-1} (I - H_k^T K_{k,0}) F_k^T \\ P_{k+1,k-1|k} &= P_{k,k-1|k-1} (I - H_k^T K_{k,0}) F_k^T \\ &\vdots = \vdots \\ P_{k+1,k-N|k} &= P_{k,k-N|k-1} (I - H_k^T K_{k,0}) F_k^T \end{aligned} \quad (15.140)$$

and the following recursions for the diagonal of the covariance matrix

$$\begin{aligned}
 P_{k|k} &= P_{k|k-1} - P_{k|k-1}H_k^T K_{k,0}^T \\
 P_{k-1|k} &= P_{k-1|k-1} - P_{k|k-1}H_k^T K_{k,1}^T \\
 P_{k-2|k} &= P_{k-2|k-1} - P_{k,k-2|k-1}H_k^T K_{k,1}^T \\
 &\vdots = \vdots \\
 P_{k-N|k} &= P_{k-N|k-1} - P_{k,k-N|k-1}H_k^T K_{k,N}^T
 \end{aligned} \tag{15.141}$$

Thus, the **FL-KS prediction step** predicts the state estimate, the state covariance, and the output using the input and state-space model parameters as

$$\begin{aligned}
 K_{k,0} &= P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1} \\
 \hat{x}_{k+1|k} &= F_k \hat{x}_{k|k-1} + G_k \vec{u}_k + F_k K_{k,0} (\vec{y}_k - H_k \hat{x}_{k|k-1}) \\
 P_{k+1|k} &= F_k P_{k|k-1}(I - H_k^T K_{k,0}^T)F_k^T + Q_k
 \end{aligned} \tag{15.142}$$

and the **FL-KS smoothing step** updates the state estimate and the state covariance using the prior state estimate and covariance at time step k for $i = i, \dots, N+1$ as

$$\begin{aligned}
 K_{k,i} &= P_{k|k-1}H_k^T(H_kP_{k|k-1}H_k^T + R_k)^{-1} \\
 \hat{x}_{k+1-i|k} &= \hat{x}_{k+2-i|k} + K_{k,i}(\vec{y}_k - \hat{y}_k) \\
 P_{k+1-i|k} &= P_{k-i|k-1} - P_{k+1,k-i|k-1}H_k^T K_{k,i}^T F_k^T \\
 P_{k+1,k-i|k} &= P_{k,k-1|k-1}(I - H_k^T K_{k,0})F_k^T
 \end{aligned} \tag{15.143}$$

where $P_{k,k|k-1} = P_{k|k-1}$ for ease of notation and, notably, the first three equations for $i = 1$ forms the standard KF correction step at time step k .

References

For more information, please refer to the following

- Simon, D., “6.1 Sequential Kalman filtering,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 150-155
- Simon, D., “6.2 Information filtering,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 156-158
- Simon, D., “6.3 Square root filtering,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 159-173
- Simon, D., “9.3 Fixed-Lag Smoothing,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 274-279

15.4 Generalized Noise for Discrete-Time Kalman Filtering

Recall that in the previous discrete-time Kalman filter recursion, one assumes two key aspects about the AWGN. The first is that the process and measurement noise are uncorrelated with each other. The second is that the process and measurement noise are uncorrelated in time, i.e., white noise. If either of these assumptions are untrue, i.e., correlated-noise or colored noise, then the standard discrete-time Kalman filter recursion must be altered for these general noise cases which are discussed in the following subsections using a least-squares approach to the estimator derivations which considers a linear stochastic state-space model as

$$\begin{aligned}\vec{x}_k &= F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} \\ \vec{y}_k &= H_k \vec{x}_k + \vec{v}_k\end{aligned}\tag{15.144}$$

with $\vec{W}_k \sim \mathcal{N}(0, Q_k)$ and $\vec{V}_k \sim \mathcal{N}(0, R_k)$.

Discrete-Time Correlated-Noise Kalman Filtering

Consider the case where the process and measurement noise are correlated, but without colored noise, i.e.,

$$\begin{aligned}\mathbb{E}[\vec{w}_k \vec{w}_j^T] &= Q_k \delta_{k-j} \\ \mathbb{E}[\vec{v}_k \vec{v}_j^T] &= R_k \delta_{k-j} \\ \mathbb{E}[\vec{w}_k \vec{v}_j^T] &= N_k \delta_{k-j+1}\end{aligned}\tag{15.145}$$

where N_k is the **noise cross-covariance** and models the expected correlation/covariance between the zero-mean noise vectors. To derive the correlated-noise Kalman filter (CN-KF), assume the same form for the prior and posterior state estimates

$$\begin{aligned}\hat{\vec{x}}_{k|k-1} &= F_{k-1} \hat{\vec{x}}_{k-1|k-1} + G_{k-1} \vec{u}_{k-1} \\ \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k (\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1})\end{aligned}\tag{15.146}$$

where the Kalman gain will be different than the standard KF as well as the posterior state covariance.

The prior state covariance by definition is

$$P_{k|k-1} = \mathbb{E} \left[\left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} \right) \left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} \right)^T \right]\tag{15.147}$$

By substitution for the linear state process and prior state estimate, one has

$$\begin{aligned}P_{k|k-1} &= \mathbb{E} \left[\left(F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} - F_{k-1} \hat{\vec{x}}_{k-1|k-1} - G_{k-1} \vec{u}_{k-1} \right) \right. \\ &\quad \left. \left(F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} - F_{k-1} \hat{\vec{x}}_{k-1|k-1} - G_{k-1} \vec{u}_{k-1} \right)^T \right]\end{aligned}\tag{15.148}$$

$$\begin{aligned} P_{k|k-1} = & \mathbb{E} \left[F_{k-1} (\vec{x}_k - \hat{\vec{x}}_{k-1|k-1}) (\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})^T F_{k-1}^T + \vec{w}_{k-1} \vec{w}_{k-1}^T \right. \\ & \left. + F_{k-1} (\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) \vec{w}_{k-1}^T + \vec{w}_{k-1} (\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})^T F_{k-1}^T \right] \end{aligned} \quad (15.149)$$

By substitution of the posterior state covariance, process noise covariance, and that recalling that $(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})$ is uncorrelated with zero-mean \vec{w}_{k-1} , one has

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \quad (15.150)$$

The posterior state covariance by definition is

$$P_{k|k} = \mathbb{E} \left[(\vec{x}_k - \hat{\vec{x}}_{k|k}) (\vec{x}_k - \hat{\vec{x}}_{k|k})^T \right] \quad (15.151)$$

By substitution for the posterior state estimate as

$$\begin{aligned} P_{k|k} = & \mathbb{E} \left[\left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} - K_k (\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}) \right) \right. \\ & \left. \left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} - K_k (\vec{y}_k - H_k \hat{\vec{x}}_{k|k-1}) \right)^T \right] \end{aligned} \quad (15.152)$$

and substitution for the measurement, one has

$$\begin{aligned} P_{k|k} = & \mathbb{E} \left[\left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} - K_k (H_k \vec{x}_k + \vec{v}_k - H_k \hat{\vec{x}}_{k|k-1}) \right) \right. \\ & \left. \left(\vec{x}_k - \hat{\vec{x}}_{k|k-1} - K_k (H_k \vec{x}_k + \vec{v}_k - H_k \hat{\vec{x}}_{k|k-1}) \right)^T \right] \end{aligned} \quad (15.153)$$

$$\begin{aligned} P_{k|k} = & \mathbb{E} \left[\left((I - K_k H_k) (\vec{x}_k - \hat{\vec{x}}_{k|k-1}) - K_k \vec{v}_k \right) \right. \\ & \left. \left((I - K_k H_k) (\vec{x}_k - \hat{\vec{x}}_{k|k-1}) - K_k \vec{v}_k \right)^T \right] \end{aligned} \quad (15.154)$$

$$\begin{aligned} P_{k|k} = & \mathbb{E} \left[(I - K_k H_k) (\vec{x}_k - \hat{\vec{x}}_{k|k-1}) (\vec{x}_k - \hat{\vec{x}}_{k|k-1})^T (I - K_k H_k)^T + K_k \vec{v}_k \vec{v}_k^T K_k^T \right. \\ & \left. - K_k \vec{v}_k (\vec{x}_k - \hat{\vec{x}}_{k|k-1})^T (I - K_k H_k)^T - (I - K_k H_k) (\vec{x}_k - \hat{\vec{x}}_{k|k-1}) \vec{v}_k^T K_k^T \right] \end{aligned} \quad (15.155)$$

By substitution for the prior covariance, measurement noise covariance, and the linear state process model from \vec{x}_{k-1} to \vec{x}_k , one has

$$\begin{aligned} P_{k|k} = & (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T \\ & + K_k \mathbb{E} \left[\vec{v}_k (F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} - \hat{\vec{x}}_{k|k-1})^T \right] (I - K_k H_k)^T \\ & - (I - K_k H_k) \mathbb{E} \left[(F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} - \hat{\vec{x}}_{k|k-1}) \vec{v}_k^T \right] K^T \end{aligned} \quad (15.156)$$

Then, by the definition of the noise cross-covariance and using the fact that \vec{x}_{k-1} , \vec{u}_{k-1} , and $\hat{\vec{x}}_{k|k-1}$ are uncorrelated with zero-mean \vec{v}_{k-1} , one has

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T - K_k N_k^T (I - K_k H_k)^T - (I - K_k H_k) N_k K_k^T \quad (15.157)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} (I - K_k H_k)^T + K_k R_k K_k^T - K_k N_k^T - N_k K_k^T + K_k (N_k^T H_k^T + H_k N_k) K_k^T \quad (15.158)$$

Thus, for the variance-minimizing Kalman gain, one must minimize the trace of the posterior state covariance. This occurs for the matrix derivative

$$\frac{\partial \text{Tr}(P_{k|k})}{\partial K_k} = -2(I - K_k H_k) P_{k|k-1} H_k^T + 2K_k R_k + 2K_k (N_k^T H_k^T + H_k N_k) - 2N_k \quad (15.159)$$

$$\frac{\partial \text{Tr}(P_{k|k})}{\partial K_k} = 2 \left[K_k \left(H_k P_{k|k-1} H_k^T + H_k N_k + N_k^T H_k^T + R_k \right) - (P_{k|k-1} H_k^T + N_k) \right] \quad (15.160)$$

Thus, to make this zero, one requires the correlated process and measurement noise Kalman gain to be

$$K_k = (P_{k|k-1} H_k^T + N_k) \left(H_k P_{k|k-1} H_k^T + H_k N_k + N_k^T H_k^T + R_k \right)^{-1} \quad (15.161)$$

The posterior state covariance can be rearranged as

$$\begin{aligned} P_{k|k} = & (I - K_k H_k) P_{k|k-1} - P_{k|k-1} H_k^T K_k^T - K_k N_k^T - N_k K_k^T \\ & + K_k (H_k P_{k|k-1} H_k^T + N_k^T H_k^T + H_k N_k + R_k) K_k^T \end{aligned} \quad (15.162)$$

Then, substituting for K_k into the first term on the second line, one has

$$\begin{aligned} P_{k|k} = & (I - K_k H_k) P_{k|k-1} - K_k N_k^T - (P_{k|k-1} H_k^T + N_k) K_k^T \\ & + (P_{k|k-1} H_k^T + N_k) \left(H_k P_{k|k-1} H_k^T + H_k N_k + N_k^T H_k^T + R_k \right)^{-1} \\ & (H_k P_{k|k-1} H_k^T + N_k^T H_k^T + H_k N_k + R_k) K_k^T \end{aligned} \quad (15.163)$$

$$\begin{aligned} P_{k|k} = & (I - K_k H_k) P_{k|k-1} - K_k N_k^T - (P_{k|k-1} H_k^T + N_k) K_k^T \\ & + (P_{k|k-1} H_k^T + N_k) \left(H_k P_{k|k-1} H_k^T + H_k N_k + N_k^T H_k^T + R_k \right)^{-1} \\ & (H_k P_{k|k-1} H_k^T + N_k^T H_k^T + H_k N_k + R_k) K_k^T \end{aligned} \quad (15.164)$$

$$\begin{aligned} P_{k|k} = & (I - K_k H_k) P_{k|k-1} - K_k N_k^T - (P_{k|k-1} H_k^T + N_k) K_k^T \\ & + (P_{k|k-1} H_k^T + N_k) K_k^T \end{aligned} \quad (15.165)$$

$$P_{k|k} = (I - K_k H_k) P_{k|k-1} - K_k N_k^T \quad (15.166)$$

Similarly, the innovation covariance can be computed as

$$S_k = \mathbb{E} [(\vec{y}_k - \hat{\vec{y}}_k)(\vec{y}_k - \hat{\vec{y}}_k)^T] \quad (15.167)$$

$$S_k = \mathbb{E} [(H_k \vec{x}_k + \vec{v}_k - H_k \hat{\vec{x}}_{k|k-1})(H_k \vec{x}_k + \vec{v}_k - H_k \hat{\vec{x}}_{k|k-1})^T] \quad (15.168)$$

$$\begin{aligned} S_k = & \mathbb{E} \left[(H_k (\vec{x}_k - \hat{\vec{x}}_{k|k-1})(\vec{x}_k - \hat{\vec{x}}_{k|k-1})^T H_k^T + \vec{v}_k \vec{v}_k^T \right. \\ & \left. + \vec{v}_k (\vec{x}_k^T - \hat{\vec{x}}_{k|k-1}^T) H_k^T + H_k (\vec{x}_k - \hat{\vec{x}}_{k|k-1}) \vec{v}_k^T \right] \end{aligned} \quad (15.169)$$

By substitution for the prior covariance, measurement noise covariance, and the linear state process model from \vec{x}_{k-1} to \vec{x}_k , one has

$$\begin{aligned} S_k = & H_k P_{k|k-1} H_k^T + R_k \\ & + \mathbb{E} \left[\vec{v}_k (\vec{x}_{k-1}^T F_{k-1}^T + \vec{u}_{k-1}^T G_{k-1}^T + \vec{w}_{k-1}^T - \hat{\vec{x}}_{k|k-1}^T) H_k^T \right. \\ & \left. + H_k (F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} - \hat{\vec{x}}_{k|k-1}) \vec{v}_k^T \right] \end{aligned} \quad (15.170)$$

Then, by the definition of the noise cross-covariance and using the fact that \vec{x}_{k-1} , \vec{u}_{k-1} , and $\hat{\vec{x}}_{k|k-1}$ are uncorrelated with zero-mean \vec{v}_{k-1} , one has

$$S_k = H_k P_{k|k-1} H_k^T + H_k N_k + N_k^T H_k^T + R_k \quad (15.171)$$

Thus, the **correlated-noise Kalman filter (CN-KF)** has a modified correction step

$$\begin{aligned} \tilde{y}_k &= \vec{y}_k - \hat{\vec{y}}_k \\ S_k &= H_k P_{k|k-1} H_k^T + H_k N_k + N_k^T H_k^T + R_k \\ K_k &= (P_{k|k-1} H_k^T + N_k) S_k^{-1} \\ \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k \tilde{y}_k \\ \hat{P}_{k|k} &= (I - K_k H_k) P_{k|k-1} - K_k N_k^T \end{aligned} \quad (15.172)$$

which simply uses the modeled cross-covariance in the innovation covariance, Kalman gain, and posterior covariance recursions.

However, in some filtering problems, one may not be able to identify the cross-covariance between the information to be fused. One common method for fusing this information is known as **covariance intersection (CI)**. Here, consider information \vec{X}_1 with mean $\hat{\vec{x}}_1$ and information matrix P_1^{-1} to be fused with information \vec{X}_2 with mean $\hat{\vec{x}}_2$ and information matrix P_2^{-1} , to form the fused information \vec{X}_ω with

mean $\hat{\vec{x}}_\omega$ and information matrix P_ω^{-1} . If the cross-covariance between \vec{X}_1 and \vec{X}_2 is known as P_{12} , one has a mean and covariance update based on a generalization of the CN-KF as

$$\begin{aligned}\hat{\vec{x}}_\omega &= K_1 \hat{\vec{x}}_1 + K_2 \hat{\vec{x}}_2 \\ P_\omega &= K_1 P_1 K_1^T + K_2 P_2 K_2^T + K_1 P_{12} K_2^T + K_2 P_{12}^T K_1^T\end{aligned}\quad (15.173)$$

However, if P_{12} is unknown, the **covariance intersection (CI) algorithm** can be formed as a convex combination of the individual information matrices, i.e.,

$$\begin{aligned}P_\omega &= \left(\omega P_1^{-1} + (1 - \omega) P_2^{-1} \right)^{-1} \\ \hat{\vec{x}}_\omega &= P_\omega \left(\omega P_1^{-1} \hat{\vec{x}}_1 + (1 - \omega) P_2^{-1} \hat{\vec{x}}_2 \right)\end{aligned}\quad (15.174)$$

where $\omega \in [0, 1]$ is a scalar optimization parameter that is chosen to minimize $\text{Tr}(P_\omega)$. The term “covariance intersection” derives from the geometrical perspective for the two-dimensional fusion case as the fused covariance ellipse is guaranteed to enclose the intersection of the two covariance ellipses with a co-located center for *any* possible cross-covariance. Furthermore, it can be shown that the CI algorithm guarantees a consistent estimate for all valid cross-covariances, i.e.,

$$P_\omega - \mathbb{E} [(\vec{x} - \hat{\vec{x}}_\omega)(\vec{x} - \hat{\vec{x}}_\omega)^T] \geq 0 \quad (15.175)$$

With this algorithm in mind, the Kalman filter can be modified for *unknown* correlation between the process and measurement noise. Thus, the **covariance intersection Kalman filter (CI-KF)** has a modified correction step

$$\begin{aligned}\tilde{y}_k &= \vec{y}_k - \hat{\vec{y}}_k \\ \omega_{opt} &= \underset{\omega \in [0, 1]}{\operatorname{argmin}} \text{Tr} \left(\omega P_{k|k-1}^{-1} + (1 - \omega) H_k^T R^{-1} H_k \right)^{-1} \\ \hat{P}_{k|k} &= \left(\omega_{opt} P_{k|k-1}^{-1} + (1 - \omega_{opt}) H_k^T R^{-1} H_k \right)^{-1} \\ K_k &= (1 - \omega_{opt}) \hat{P}_{k|k} H_k^T R^{-1} \\ \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k \tilde{y}_k\end{aligned}\quad (15.176)$$

where one can use any scalar search optimization algorithm to find ω_{opt} . Notably, this CI concept can be extended to the **generalized covariance intersection (GCI) algorithm** which fuses N information, $\vec{X}_1, \dots, \vec{X}_N$, with unknown correlation between all information with the possible consideration of known bounds of the correlation factors.

Discrete-Time Colored-Noise Kalman Filtering

Consider the case where the process noise is uncorrelated with the measurement noise, but is colored, i.e.,

$$\begin{aligned}\mathbb{E} [\vec{w}_k \vec{w}_k^T] &= Q_k \\ \mathbb{E} [\vec{w}_k \vec{w}_{k-1}^T] &= \theta_{k-1} Q_k\end{aligned}\quad (15.177)$$

In this case, one can model the state process as

$$\vec{x}_k = F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} \quad (15.178)$$

and the process noise by an additional n_x^{th} -order Markov process

$$\vec{w}_k = \theta_{k-1} \vec{w}_{k-1} + \vec{\zeta}_{k-1} \quad (15.179)$$

where $\vec{\zeta}_{k-1}$ is zero-mean white noise with covariance C_ζ that is uncorrelated with \vec{w}_{k-1} .

This construction provides the augmented state process equation as

$$\begin{aligned} \begin{bmatrix} \vec{x}_k \\ \vec{w}_k \end{bmatrix} &= \begin{bmatrix} F_{k-1} & I \\ 0 & \theta_{k-1} \end{bmatrix} \begin{bmatrix} \vec{x}_{k-1} \\ \vec{w}_{k-1} \end{bmatrix} + \begin{bmatrix} G_{k-1} \\ 0 \end{bmatrix} \vec{u}_{k-1} + \begin{bmatrix} 0 \\ \vec{\zeta}_{k-1} \end{bmatrix} \\ \vec{x}_k^* &= F_{k-1}^* \vec{x}_{k-1}^* + G_{k-1}^* \vec{u}_{k-1} + \vec{w}_{k-1}^* \end{aligned} \quad (15.180)$$

which has augmented process noise covariance as

$$Q_{k-1}^* = \begin{bmatrix} 0 & 0 \\ 0 & C_\zeta \end{bmatrix} \quad (15.181)$$

Next, consider the case where the measurement noise is uncorrelated with the process noise, but is colored, i.e.,

$$\begin{aligned} \mathbb{E} [\vec{v}_k \vec{v}_k^T] &= R_k \\ \mathbb{E} [\vec{v}_k \vec{v}_{k-1}^T] &= \theta_{k-1} R_k \end{aligned} \quad (15.182)$$

In this case, one can model the measurement equation as

$$\vec{y}_k = H_k \vec{x}_k + \vec{v}_k \quad (15.183)$$

and the measurement noise by a n_y^{th} -order Markov process

$$\vec{v}_k = \theta_{k-1} \vec{v}_{k-1} + \vec{\zeta}_{k-1} \quad (15.184)$$

where $\vec{\zeta}_{k-1}$ is zero-mean white noise with covariance C_ζ that is uncorrelated with \vec{v}_{k-1} .

While one can use the same augmented state approach, this results in a singular measurement noise covariance which can cause numerical difficulties. Thus, it is preferable to use measurement differencing as

$$\Delta \vec{y}_{k+1,k} = \vec{y}_{k+1} - \theta_k \vec{y}_k \quad (15.185)$$

By substitution for the measurement equation, process equation, and measurement noise, one has

$$\Delta \vec{y}_{k+1,k} = H_{k+1} (F_k \vec{x}_k + G_k \vec{u}_k + \vec{w}_k) + \vec{v}_{k-1} - \theta_k H_k \vec{x}_k - \theta_k \vec{v}_{k-1} \quad (15.186)$$

Rearranging, one has

$$\Delta \vec{y}_{k+1,k} = H_{k+1} (F_k - \theta_k H_k + \vec{w}_k) \vec{x}_k + H_{k+1} G_k \vec{u}_k + H_{k+1} \vec{w}_k + \vec{\zeta}_k \quad (15.187)$$

which provides the alternative measurement model

$$\vec{y}_k^* = \Delta \vec{y}_{k+1,k} - H_{k+1} G_k \vec{u}_k = H_k^* \vec{x}_k + \vec{v}_k^* \quad (15.188)$$

with

$$H_k^* = H_{k+1} F_k - \theta_k H_k \quad (15.189)$$

and

$$\vec{v}_k^* = H_{k+1} \vec{w}_k + \vec{\zeta}_k \quad (15.190)$$

where the alternative noise covariance is

$$R_k^* = \mathbb{E} [\vec{v}_k^* \vec{v}_k^{*T}] = H_{k+1} Q_k H_{k+1}^T + C_\zeta \quad (15.191)$$

and alternative noise cross-covariance is

$$N_k^* = \mathbb{E} [\vec{w}_k \vec{v}_k^{*T}] = Q_k H_{k+1}^T \quad (15.192)$$

Thus, one can utilize the correlated-noise Kalman filter recursion to derive the one-step **colored-measurement-noise Kalman filter (CMN-KF)** as

$$\begin{aligned} K_k &= P_{k|1:k} H_k^{*T} (H_k^* P_{k|1:k} H_k^{*T} + R_k^*)^{-1} \\ \hat{x}_{k|1:k+1} &= \hat{x}_{k|1:k} + K_k (\vec{y}_k^* - H_k^* \hat{x}_{k|1:k}) \\ P_{k|1:k+1} &= (I - K_k H_k^*) P_{k|1:k} (I - K_k H_k^*) + K_k R_k^* K_k^T \\ C_k &= Q_k H_{k+1}^T (H_k^* P_{k|1:k} H_k^{*T} + R_k^*)^{-1} \\ \hat{x}_{k+1|1:k+1} &= F_k \hat{x}_{k|1:k+1} + C_k (\vec{y}_k^* - H_k^* \hat{x}_{k|1:k+1}) \\ P_{k+1|1:k+1} &= F_k P_{k|1:k+1} F_k^T + Q_k - C_k H_{k+1} Q_k - F_k K_k Q_k H_{k+1}^T - H_{k+1} Q_k K_k^T F_k \end{aligned} \quad (15.193)$$

References

For more information, please refer to the following

- Julier, S., and Uhlmann, J. “12 General Decentralized Data Fusion with Covariance Intersection (CI),” in *Handbook of multisensor data fusion*, CRC Press LLC, 2001, pp. 12-1 to 12-25
- Reece, S., and Roberts, S., “Robust, low-bandwidth, multi-vehicle mapping,” in *8th International Conference on Information Fusion*, Vol 2, 2005, pp. 8
- Reinhardt, M., Kulkarni, S., and Hanebeck, U. D., “Generalized Covariance Intersection based on Noise Decomposition,” in *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*, 2014, pp. 1-8
- Simon, D., “7.1 Correlated Process and Measurement Noise,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 184-188
- Simon, D., “7.2 Colored Process and Measurement Noise,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 188-193
- Simon, D., “7.3 Steady-State Filtering,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 193-195

15.5 Discretization and Continuous-Time Kalman Filtering

Discretized Stochastic State-Space Model

Consider a discrete-time process model with identity state transition matrix, $F_k = I$, as

$$\vec{x}_k = \vec{x}_{k-1} + \vec{w}_{k-1} \quad (15.194)$$

with \vec{W}_k as a discrete-time white noise process, i.e., zero-mean with covariance Q_{d-t} , and initial condition $\vec{x}_0 = 0$ which has the solution

$$\vec{x}_k = \vec{w}_0 + \cdots + \vec{w}_k \quad (15.195)$$

which is known as a **random walk**. If $\vec{W}_k \sim \mathcal{N}(0, Q_{d-t})$, then this is a **Gaussian random walk**.

A random walk has a mean given by

$$\mathbb{E}[\vec{x}_k] = \mathbb{E}[\vec{w}_0 + \cdots + \vec{w}_k] = 0 \quad (15.196)$$

and a covariance given by

$$\mathbb{E}[\vec{x}_k \vec{x}_k^T] = \mathbb{E}[(\vec{w}_0 + \cdots + \vec{w}_k)(\vec{w}_0 + \cdots + \vec{w}_k)^T] \quad (15.197)$$

$$\mathbb{E}[\vec{x}_k \vec{x}_k^T] = \mathbb{E}[\vec{w}_0 \vec{w}_0^T] + \cdots + \mathbb{E}[\vec{w}_k \vec{w}_k^T] \quad (15.198)$$

$$\mathbb{E}[\vec{x}_k \vec{x}_k^T] = kQ_{d-t} \quad (15.199)$$

Next, consider the continuous-time equivalence with $t = k\Delta t$ which provides a covariance

$$\mathbb{E}[\vec{x}(t) \vec{x}^T(t)] = \mathbb{E}[\vec{x}_k \vec{x}_k^T] = kQ_{d-t} \quad (15.200)$$

Thus, if one considers $\vec{X}(t)$ as the analogous continuous-time process represented as the stochastic differential equation

$$\dot{\vec{x}}(t) = \vec{w}(t) \quad (15.201)$$

with $\vec{w}(t)$ as a continuous-time white noise with mean function given by

$$\mathbb{E}[\vec{w}(t)] = 0 \quad (15.202)$$

and an auto-covariance function given by

$$\mathbb{E}[\vec{w}(t) \vec{w}^T(\tau)] = \frac{Q_{d-t}}{\Delta t} \delta(t - \tau) = Q_{c-t} \delta(t - \tau) \quad (15.203)$$

where $\delta(t - \tau)$ is the Dirac delta function, Q_{c-t} is the continuous-time process noise covariance, and $Q_{d-t} = Q_{c-t}\Delta t$ is the **discretized process noise covariance**. Thus, $\vec{W}(t)$ is infinitely correlated with itself at $t = \tau$ as defined previously. Note that if \vec{X}_k is a Gaussian random walk, then $\vec{X}(t)$ is a Wiener process of its limit as the sampling interval, Δt , approaches zero.

One can show this is the continuous-time equivalence of the discrete-time random walk by assessing the covariance of $\vec{X}(t)$, i.e.,

$$\mathbb{E}[\vec{x}(t) \vec{x}^T(t)] = \mathbb{E} \left[\int_0^t \dot{\vec{x}}(\zeta) d\zeta \int_0^t \dot{\vec{x}}^T(\tau) d\tau \right] \quad (15.204)$$

$$\mathbb{E}[\vec{x}(t)\vec{x}^T(t)] = \int_0^t \int_0^t \mathbb{E}[\vec{w}(\zeta)\vec{w}^T(\tau)] d\zeta d\tau \quad (15.205)$$

$$\mathbb{E}[\vec{x}(t)\vec{x}^T(t)] = \int_0^t \int_0^t \frac{Q_{d-t}}{\Delta t} \delta(\zeta - \tau) d\zeta d\tau \quad (15.206)$$

$$\mathbb{E}[\vec{x}(t)\vec{x}^T(t)] = \int_0^t \frac{Q_{d-t}}{\Delta t} d\tau \quad (15.207)$$

$$\mathbb{E}[\vec{x}(t)\vec{x}^T(t)] = \frac{Q_{d-t}t}{\Delta t} \quad (15.208)$$

$$\mathbb{E}[\vec{x}(t)\vec{x}^T(t)] = kQ_{d-t} \quad (15.209)$$

Next, consider a discrete-time hidden constant-state model with an identity state transition matrix $F_k = I$, no process noise with $Q_k = 0$, and an identity measurement matrix, $H_k = I$ as

$$\begin{aligned} \vec{x}_k &= \vec{x}_{k-1} \\ \vec{y}_k &= \vec{x}_k + \vec{v}_k \end{aligned} \quad (15.210)$$

with \vec{V}_k as a discrete-time white noise process, i.e., zero-mean with covariance R_{d-t} . For the Kalman filter, the one-step posterior state covariance, one has

$$\begin{aligned} K_k &= P_{k-1|k-1} (P_{k-1|k-1} + R_{d-t})^{-1} \\ P_{k|k} &= (I - K_k) P_{k-1|k-1} \end{aligned} \quad (15.211)$$

Combining, one has

$$P_{k|k} = (P_{k-1|k-1} + R_{d-t} - P_{k-1|k-1}) (P_{k-1|k-1} + R_{d-t})^{-1} P_{k-1|k-1} \quad (15.212)$$

$$P_{k|k} = R (P_{k-1|k-1} + R_{d-t})^{-1} P_{k-1|k-1} \quad (15.213)$$

or, beginning from P_0 at $k = 0$, one has

$$P_{k|k} = R (kP_0 + R_{d-t})^{-1} P_0 \quad (15.214)$$

$$\lim_{P_0 \rightarrow \infty} P_{k|k} = \frac{R_{d-t}}{k} = \frac{R_{d-t}\Delta t}{t_k} \quad (15.215)$$

where the state covariance will be independent of the sample time for some constant continuous-time measurement noise covariance,

$$R_{c-t} = R_{d-t}\Delta t \quad (15.216)$$

This implies the analogous continuous-time process represented as the algebraic measurement equation

$$\vec{y}(t) = \vec{x}(t) + \vec{v}(t) \quad (15.217)$$

with $\vec{v}(t)$ as a continuous-time white noise process with mean function given by

$$\mathbb{E}[\vec{v}(t)] = 0 \quad (15.218)$$

and an auto-covariance function given by

$$\mathbb{E}[\vec{v}(t)\vec{v}^T(\tau)] = R_{d-t}\Delta t\delta(t-\tau) = R_{c-t}\delta(t-\tau) \quad (15.219)$$

where $\delta(t-\tau)$ is the Dirac delta function, R_{c-t} is the continuous-time process noise covariance, and $R_{d-t} = R_{c-t}/\Delta t$ is the **discretized measurement noise covariance**.

Finally, consider a continuous-time LTI stochastic state-space model

$$\begin{aligned}\dot{\vec{x}}(t) &= A\vec{x}(t) + B\vec{u}(t) + \vec{w}(t) \\ \vec{y}(t) &= C\vec{x}(t) + \vec{v}(t)\end{aligned} \quad (15.220)$$

with zero-mean process noise $\vec{W}(t)$ and covariance, Q_{c-t} , and zero-mean measurement noise $\vec{V}(t)$ and covariance, R_{c-t} . One can compute the **discretized LTI stochastic state-space model** with time interval $\Delta t = t/k$ as

$$\begin{aligned}\dot{\vec{x}}_k &= F\vec{x}_{k-1} + G\vec{u}_{k-1} + \vec{w}_{k-1} \\ \vec{y}_k &= H\vec{x}_k + \vec{v}_k\end{aligned} \quad (15.221)$$

with zero-mean process noise \vec{W}_k and covariance, $Q_{d-t} = Q_{c-t}\Delta t$, zero-mean measurement noise \vec{V}_k and covariance, $R_{d-t} = R_{c-t}/\Delta t$, discretized state transition matrix

$$F = \exp(A\Delta t) \quad (15.222)$$

discretized input matrix

$$G = \exp(A\Delta t) \int_0^{\Delta t} \exp(-A\tau) d\tau B \quad (15.223)$$

or if A^{-1} exists, one has

$$G = \exp(A\Delta t) (I - \exp(-A\Delta t)) A^{-1} B = (\exp(A\Delta t) - I) A^{-1} B \quad (15.224)$$

and measurement matrix

$$H = C \quad (15.225)$$

Furthermore, for $\Delta t \ll 1$, one has the first-order approximation

$$F \approx I + A\Delta t \quad (15.226)$$

$$G \approx B\Delta t \quad (15.227)$$

Continuous-Time Kalman Filtering

With the discretized stochastic state-space model in mind, consider the discrete-time Kalman gain as

$$K_k = P_{k|k-1} H^T (H P_{k|k-1} H^T + R)^{-1} \quad (15.228)$$

By substitution for the discretized stochastic state-space model with the first-order approximation, one has

$$K_k = P_{k|k-1} C^T \left(C P_{k|k-1} C^T + \frac{R_{c-t}}{\Delta t} \right)^{-1} \quad (15.229)$$

$$\frac{K_k}{\Delta t} = P_{k|k-1} C^T \left(C P_{k|k-1} C^T \Delta t + R_{c-t} \right)^{-1} \quad (15.230)$$

$$\lim_{\Delta t \rightarrow 0} \frac{K_k}{\Delta t} = P_{k|k-1} C^T R_{c-t}^{-1} \quad (15.231)$$

Next, consider the one-step prior discrete-time Kalman filter for the state covariance

$$P_{k+1|k} = F(I - K_k H)P_{k|k-1}F^T + Q \quad (15.232)$$

By substitution for the discretized stochastic state-space model with the first-order approximation, one has

$$P_{k+1|k} = (I + A\Delta t)(I - K_k C)P_{k|k-1}(I + A\Delta t)^T + Q_{c-t}\Delta t \quad (15.233)$$

$$\begin{aligned} P_{k+1|k} = & P_{k|k-1} + AP_{k|k-1}\Delta t + P_{k|k-1}A^T\Delta t + AP_{k|k-1}A^T(\Delta t)^2 \\ & - K_k CP_{k|k-1} - AK_k CP_{k|k-1}\Delta t - K_k CP_{k|k-1}A^T\Delta t - AK_k CP_{k|k-1}A^T(\Delta t)^2 + Q_{c-t}\Delta t \end{aligned} \quad (15.234)$$

Subtracting $P_{k|k-1}$ from both sides and dividing by Δt , one has

$$\begin{aligned} \frac{P_{k+1|k} - P_{k|k-1}}{\Delta t} = & AP_{k|k-1} + P_{k|k-1}A^T + AP_{k|k-1}A^T\Delta t \\ & - \frac{K_k CP_{k|k-1}}{\Delta t} - AK_k CP_{k|k-1} - K_k CP_{k|k-1}A^T - AK_k CP_{k|k-1}A^T\Delta t + Q_{c-t} \end{aligned} \quad (15.235)$$

Taking the limit with $\lim_{\Delta t \rightarrow 0} K_k(\Delta t)^{-1} = P_{k|k-1}C^T R_{c-t}^{-1}$ in mind, one has

$$\dot{P} = \lim_{\Delta t \rightarrow 0} \frac{P_{k+1|k} - P_{k|k-1}}{\Delta t} = AP + PA^T - PC^T R_{c-t}^{-1} CP + Q_{c-t} \quad (15.236)$$

which is the forwards Riccati difference equation for the LQE and

$$K(t) = PC^T R_{c-t}^{-1} \quad (15.237)$$

Note that integrating this equation for a solution to $P(t)$ requires only $n_x(n_x + 1)/2$ integrations as P is symmetric.

Finally, consider the one-step prior discrete-time Kalman filter for the state estimate

$$\hat{x}_{k|k} = F\hat{x}_{k-1|k-1} + G\vec{u}_{k-1} + K_k \left(\vec{y}_k - H \left(F\hat{x}_{k-1|k-1} + G\vec{u}_{k-1} \right) \right) \quad (15.238)$$

By substitution for the discretized stochastic state-space model with the first-order approximation, one has

$$\begin{aligned} \hat{x}_{k|k} = & (I + A\Delta t)\hat{x}_{k-1|k-1} + B\Delta t\vec{u}_{k-1} \\ & + K_k \left(\vec{y}_k - C \left(I + A\Delta t \right) \hat{x}_{k-1|k-1} + B\Delta t\vec{u}_{k-1} \right) \end{aligned} \quad (15.239)$$

$$\begin{aligned} \hat{x}_{k|k} = & \hat{x}_{k-1|k-1} + A\Delta t\hat{x}_{k-1|k-1} + B\Delta t\vec{u}_{k-1} \\ & + K_k \left(\vec{y}_k - C\hat{x}_{k-1|k-1} + CA\Delta t\hat{x}_{k-1|k-1} + CB\Delta t\vec{u}_{k-1} \right) \end{aligned} \quad (15.240)$$

Subtracting $x_{k-1|k-1}$ from both sides and dividing by Δt , one has

$$\begin{aligned} \frac{\hat{x}_{k|k} - \hat{x}_{k-1|k-1}}{\Delta t} &= A\hat{x}_{k-1|k-1} + B\vec{u}_{k-1} \\ &\quad + \frac{K_k}{\Delta t} \left(\vec{y}_k - C(I + A\Delta t)\hat{x}_{k-1|k-1} + B\Delta t\vec{u}_{k-1} \right) \end{aligned} \quad (15.241)$$

Taking the limit with $\lim_{\Delta t \rightarrow 0} K_k(\Delta t)^{-1} = PC^T R_{c-t}^{-1}$ in mind, one has

$$\dot{\hat{x}} = \lim_{\Delta t \rightarrow 0} \frac{\hat{x}_{k|k} - \hat{x}_{k-1|k-1}}{\Delta t} = A\hat{x}_{k-1|k-1} + B\vec{u}_{k-1} + PC^T R_{c-t}^{-1} \left(\vec{y}_k - C\hat{x} \right) \quad (15.242)$$

In summary, these limiting arguments for the discretized stochastic state-space system provide an alternative derivation of the **continuous-time Kalman filter (KF)**, also known as the **Kalman-Bucy filter (KBF)**, also known as the **continuous-time linear-quadratic estimator (LQE)**, as the vector and matrix differential equations

$$\begin{aligned} \dot{P} &= AP + PA^T + Q_{c-t} - PC^T R_{c-t}^{-1} CP \\ K &= PC^T R_{c-t}^{-1} \\ \dot{\hat{x}}(t) &= A\hat{x}(t) + B\vec{u}(t) + K(\vec{y} - C\hat{x}(t)) \end{aligned} \quad (15.243)$$

Notably, the continuous-time **steady-state Kalman filter** uses a constant Kalman gain in the filter $K_\infty = P_\infty C^T R_{c-t}^{-1}$ which solves the CARE and results in a continuous-time LTI filter. In some cases, the continuous-time Kalman filter converges to this steady-state condition. In this case, consider the Laplace transform of the continuous-time zero-input steady-state Kalman filter

$$s\hat{x}(s) = A\hat{x}(s) + K_\infty(\vec{y}(s) - C\hat{x}(s)) \quad (15.244)$$

$$(sI - A + K_\infty C)\hat{x}(s) = K_\infty \vec{y}(s) \quad (15.245)$$

$$\hat{x}(s) = (sI - A + K_\infty C)^{-1} K_\infty \vec{y}(s) \quad (15.246)$$

which is the transfer function matrix of the causal **Wiener filter**.

Continuous-Time Correlated-Noise Kalman Filtering

Similarly, consider the case where the continuous-time process and measurement noise are correlated, but without colored noise, i.e.,

$$\begin{aligned} \mathbb{E}[\vec{w}(t)\vec{w}^T(\tau)] &= Q_{c-t}\delta(t-\tau) \\ \mathbb{E}[\vec{v}(t)\vec{v}^T(\tau)] &= R_{c-t}\delta(t-\tau) \\ \mathbb{E}[\vec{w}(t)\vec{v}^T(\tau)] &= N_{c-t}\delta(t-\tau) \end{aligned} \quad (15.247)$$

where N_{c-t} is the **continuous-time noise cross-covariance** and models the expected correlation/covariance between the zero-mean noise vectors.

To derive the continuous-time correlated-noise Kalman-Bucy filter (CN-KBF), consider the following Lagrange multiplier matrix, Λ , on the measurement equation substituted into the process equation, i.e.,

$$\dot{\vec{x}} = A\vec{x} + B\vec{u} + \vec{w} + \Lambda(\vec{y} - C\vec{x} - \vec{v}) \quad (15.248)$$

Rearranging, one has

$$\dot{\vec{x}} = (A - \Lambda C)\vec{x} + B\vec{u} + \Lambda\vec{y} + \vec{w} - \Lambda\vec{v} \quad (15.249)$$

which can be defined as transformed process

$$\dot{\vec{x}} = A^*\vec{x} + \vec{u}^* + \vec{w}^* \quad (15.250)$$

with

$$A^* = A - \Lambda C \quad (15.251)$$

$$\vec{u}^* = B\vec{u} + \Lambda\vec{y} \quad (15.252)$$

and

$$\vec{w}^* = \vec{w} - \Lambda\vec{v} \quad (15.253)$$

Then, consider the cross-covariance between this transformed process noise and the measurement noise given by

$$\mathbb{E}[\vec{w}^*(t)\vec{v}^T(\tau)] = \mathbb{E}[(\vec{w}(t) - \Lambda\vec{v}(t))\vec{v}^T(\tau)] \quad (15.254)$$

$$\mathbb{E}[\vec{w}^*(t)\vec{v}^T(\tau)] = \mathbb{E}[\vec{w}(t)\vec{v}^T(\tau)] - \Lambda\mathbb{E}[\vec{v}(t)\vec{v}^T(\tau)] \quad (15.255)$$

By definition, one has

$$\mathbb{E}[\vec{w}^*(t)\vec{v}^T(\tau)] = N_{c-t} - \Lambda R_{c-t} \quad (15.256)$$

Thus, to utilize the uncorrelated KBF recursion equation, one should choose $\Lambda = N_{c-t}R_{c-t}^{-1}$.

With this choice, the transformed process noise covariance is given by

$$Q_{c-t}^* = \mathbb{E}[\vec{w}^*(t)\vec{w}^{*T}(\tau)] \quad (15.257)$$

$$Q_{c-t}^* = \mathbb{E}[(\vec{w}(t) - N_{c-t}R_{c-t}^{-1}\vec{v}(t))(\vec{w}(\tau) - N_{c-t}R_{c-t}^{-1}\vec{v}(\tau))] \quad (15.258)$$

$$Q_{c-t}^* = Q_{c-t} - N_{c-t}R_{c-t}^{-1}N_{c-t}^T - N_{c-t}R_{c-t}^{-1}N_{c-t}^T + N_{c-t}R_{c-t}^{-1}N_{c-t}^T \quad (15.259)$$

$$Q_{c-t}^* = Q_{c-t} - N_{c-t}R_{c-t}^{-1}N_{c-t}^T \quad (15.260)$$

Then, the Riccati differential equation for the state covariance can be written as

$$\dot{P} = A^*P + PA^{*T} + Q_{c-t}^* - PRC^TR_{c-t}^{-1}CP \quad (15.261)$$

By substitution, one has

$$\dot{P} = (A - N_{c-t}R_{c-t}^{-1}C)P + P(A - NR_{c-t}^{-1}C)^T + Q_{c-t} - N_{c-t}R_{c-t}^{-1}N_{c-t}^T - PC^TR_{c-t}^{-1}CP \quad (15.262)$$

Defining the transformed Kalman gain as

$$K^* = PC^TR_{c-t}^{-1} + N_{c-t}R_{c-t}^{-1} = (PC^T + N_{c-t})R_{c-t}^{-1} \quad (15.263)$$

The Riccati differential equation can be rewritten as

$$\dot{P} = AP + PA^T + Q_{c-t} - K^* R_{c-t}^{-1} K^{*T} \quad (15.264)$$

Using the transformed process equation with the standard Kalman gain, one has

$$\dot{\hat{x}} = A^* \dot{\hat{x}} + \vec{u}^* + K(\vec{y} - C \hat{x}) \quad (15.265)$$

$$\dot{\hat{x}} = (A - N_{c-t} R_{c-t}^{-1} C) \dot{\hat{x}} + (B \vec{u} + N_{c-t} R_{c-t}^{-1} \vec{y}) + PC^T R_{c-t}^{-1} (\vec{y} - C \hat{x}) \quad (15.266)$$

Rearranging, one has

$$\dot{\hat{x}} = A \dot{\hat{x}} + B \vec{u} + N_{c-t} R_{c-t}^{-1} \vec{y} - N_{c-t} R_{c-t}^{-1} C \dot{\hat{x}} + PC^T R_{c-t}^{-1} (\vec{y} - C \hat{x}) \quad (15.267)$$

or

$$\dot{\hat{x}} = A \dot{\hat{x}} + B \vec{u} + K^* (\vec{y} - C \hat{x}) \quad (15.268)$$

Thus, the **correlated-noise Kalman-Bucy filter (KBF)**, also known as the **continuous-time correlated-noise Kalman filter (CT-CN-KF)**, can be summarized as

$$\begin{aligned} \dot{P} &= AP + PA^T + Q_{c-t} - (PC^T + N_{c-t})R_{c-t}^{-1}(CP + N_{c-t}^T) \\ K &= (PC^T + N_{c-t})R_{c-t}^{-1} \\ \dot{\hat{x}}(t) &= A \hat{x}(t) + B \vec{u}(t) + K(\vec{y} - C \hat{x}(t)) \end{aligned} \quad (15.269)$$

where $\mathbb{E}[\vec{w}(t) \vec{v}^T(\tau)] = N\delta(t - \tau)$.

References

For more information, please refer to the following

- Simon, D., “8.1 Discrete-Time and Continuous-Time White Noise,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 230-233
- Simon, D., “8.2 Derivation of the Continuous-Time Kalman Filter,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 233-237
- Simon, D., “8.4 Generalizations of the Continuous-Time Filter,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 247-251

15.6 Multiple Model Filtering

Hybrid-state estimation is estimation of a state with both continuous and discrete components, as one must estimate both the continuously-valued target state and the discrete-valued motion-mode.

Multi-Model Filtering Methods

The primary approach in the presence of motion-mode uncertainty is the **multi-model (MM)** filtering method which applies to any hybrid system estimation. These methods assume a **model set**, \mathbb{M} , as possible candidates of the true mode in effect at the current time step. Thus, one can run a **elementary filter bank**, each based on a unique model $\mathcal{M}^{(i)} \in \mathbb{M}$. Thus, MM methods estimate the hybrid process based on the combined results of these elemental filters, providing an integrated approach to the joint model decision and state estimation problem, e.g. maneuvering target tracking. This is a semi-parametric approach as its model coverage is in between parametric and non-parametric approaches to this problem. This section will describe MM methods for MJLSs.

For a MJLS, the i^{th} model in the MM methods is defined as

$$\begin{aligned}\vec{x}_k &= F_{k-1}^{(i)} \vec{x}_{k-1} + G_k^{(i)} \vec{w}_{k-1} \\ \vec{y}_k &= H_k^{(i)} \vec{x}_k + \vec{v}_k^{(i)}\end{aligned}\quad (15.270)$$

where $\vec{w}_{k-1}^{(i)}$ has mean $\bar{w}_{k-1}^{(i)}$ and covariance $Q_{k-1}^{(i)}$ and $\vec{v}_k^{(i)}$ has mean $\bar{v}_k^{(i)}$ and covariance $R_k^{(i)}$.

The superscript (i) denotes quantities pertaining to model $\mathcal{M}^{(i)} \in \mathbb{M}$ and the jumps of the modal state are assumed to have homogeneous transition probabilities, i.e.

$$\Pr(\mathcal{M}_k^{(j)} | \mathcal{M}_{k-1}^{(i)}) = \pi_{ji} \quad \forall \mathcal{M}^{(i)}, \mathcal{M}^{(j)}, k \quad (15.271)$$

where $\mathcal{M}_k^{(i)}$ denotes the event that model $\mathcal{M}_k^{(i)}$ matches the modal state in effect at time step k , i.e.

$$\mathcal{M}_k^{(i)} = \{\mathcal{S}_k = \mathcal{M}^{(i)}\} \quad (15.272)$$

Similarly, the finite sequence of events are denoted as

$$\mathcal{S}_{1:k} = \{\mathcal{S}_1^{(i_1)}, \dots, \mathcal{S}_k^{(i_k)}\} \quad (15.273)$$

and

$$\mathcal{M}_{1:k} = \{\mathcal{M}_1^{(i_1)}, \dots, \mathcal{M}_k^{(i_k)}\} \quad (15.274)$$

Finally, it should be noted that the model set, \mathbb{M} , is generally a mathematical approximation of the true mode space, \mathbb{S} , and may contain fewer elements and have simpler model descriptions than the true possible set of system behaviors indicated by \mathbb{S} .

MM estimation algorithms have four key design elements. First, model set design refers to the offline selection and possible online adaptation of the model set. Second, cooperation strategy design refers to the possible pruning of unlikely model sequences, merging of “similar” model sequences, selection of the most likely model sequences, and any iterations done on these actions. Third, the filtering of the base state conditioned on some assumed mode sequence which is the only step common with non-MM algorithms. Fourth, the output process which generates overall estimates using results of all filters as well as measurements, e.g. fusing estimates, selecting the best ones. There have been three generations of MM algorithms which made different assumptions about the mode and model sets. Defining the three

A: The true modal state is time-invariant, i.e. $\mathcal{S}_k = \mathcal{S} \quad \forall k$.

B: The mode space and model set are time-variant and identical, i.e. $\mathbb{S}_k = \mathbb{S} \quad \forall k$, $\mathbb{M}_k = \mathbb{M} \quad \forall k$, $\mathbb{S} = \mathbb{M} \quad \forall k$.

C: The true mode sequence is Markov or semi-Markov.

The first generation of **autonomous MM (AMM)** assumes A and B, the second generation of **cooperating multi-model (CMM)** algorithms assumes B and C, while the third generation of **variable-structure multi-model (VSMM)** assumes only C. This section will focus on CMM algorithms which typically form the building blocks of the third generation.

Cooperating Multi-Model Filtering

In CMM, there are n_M^k possible model sequence realizations at time step k where $n_M = |\mathbb{M}|$ is the number of possible models at each k . This allows one to represent a generic realization of a model sequence through k as

$$\mathcal{M}_{1:k}^{(i_{1:k})} = \{\mathcal{M}_1^{(i_1)}, \dots, \mathcal{M}_k^{(i_k)}\} = \{\mathcal{S}_{1:k} = \mathcal{M}^{(i_{1:k})}\} \in \mathbb{M}^k \quad (15.275)$$

where \mathbb{M}^k is the set of all such sequence events. Furthermore, one can use $i_{1:k} \in \mathbb{M}_{1:k}$ to denote $\mathcal{M}_{1:k}^{(i_{1:k})} \in \mathbb{M}^k$ which has n_M^k members where \mathbb{M} is the index set of \mathbb{M} and has n_M members. Notably, a single modal state sequence is denoted

$$\mathcal{S}_{1:k} = (\mathcal{S}_1, \dots, \mathcal{S}_k) \quad (15.276)$$

a single model sequence is

$$\mathcal{M}^{(i_{1:k})} = (\mathcal{M}^{i_1}, \dots, \mathcal{M}^{i_k}), \quad \mathcal{M}^{(i_n)} \in \mathbb{M} \quad (15.277)$$

To estimate the base state of hybrid system, this section will assess two different optimal estimation schemes: MMSE-CMM and MAP-CMM. These two estimators are primarily used in CMM maneuvering target tracking due to the fact that the motion-mode is never *exactly* resolved by the model used, e.g. an aircraft never exhibits an *exactly* constant turn-rate and even if it did, the turn-rate would not be *exactly* equal to one of the model turn-rates. These two estimators are robust to this inherent model mismatch as they rely on all component densities as will be shown.

Using the law of total expectation, the **CMM-MMSE base state estimator** at time step k is given by

$$\begin{aligned} \hat{x}_{k|k}^{MMSE} &= \mathbb{E} [\vec{x}_k | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}] \\ &= \sum_{i_{1:k} \in \mathbb{M}_{1:k}} \mathbb{E} [\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_{1:k}^{(i_{1:k})}] \Pr(\mathcal{M}_{1:k}^{(i_{1:k})} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}) \\ &= \sum_{i_{1:k} \in \mathbb{M}_{1:k}} \hat{x}_{k|k}^{(i_{1:k})} \mu_{1:k}^{(i_{1:k})} \end{aligned} \quad (15.278)$$

where $\mu_{1:k}^{(i_{1:k})} = \Pr(\mathcal{M}_{1:k}^{(i_{1:k})} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C})$ is the *a posteriori* mode sequence event probability assuming the mode sequence in effect is one and only one mode, but is possibly any one of the modes in the set \mathbb{M}^k , $\hat{x}_{k|k}^{(i_{1:k})} = \mathbb{E} [\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_{1:k}^{(i_{1:k})}]$ is the conditional MMSE estimate assuming sequence $\mathcal{M}_{1:k}^{(i_{1:k})}$ is true.

Alternatively, one can write

$$\begin{aligned}\hat{x}_{k|k}^{MMSE} &= \sum_{i \in M} \mathbb{E} \left[\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_k^{(i)} \right] \Pr \left(\mathcal{M}_k^{(i)} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right) \\ &= \sum_{i \in M} \hat{x}_{k|k}^{(i)} \mu_k^{(i)}\end{aligned}\quad (15.279)$$

where $\hat{x}_{k|k}^{(i)} = \mathbb{E} \left[\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_k^{(i)} \right]$ is the conditional MMSE estimate assuming model $\mathcal{M}_k^{(i)}$ is in effect at time step k and $\mu_k^{(i)} = \Pr \left(\mathcal{M}_k^{(i)} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right)$ is a *a posteriori* mode probability that $\mathcal{M}^{(i)}$ is in effect at time step k , in effect, lumping much of $\hat{x}_{k|k}^{(i)}$ as the summation over the model sequences is the same as over the current models.

It can be shown that $\hat{x}_{k|k}^{MMSE}$ is unbiased with the minimal conditional MSE matrix, i.e. the minimum of the base-state error covariance

$$\begin{aligned}P_{k|k} &= \text{MSE} \left(\hat{x}_{k|k} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right) \\ &= \sum_{i_{1:k} \in M_{1:k}} \left[\text{MSE} \left(\hat{x}_{k|k}^{(i_{1:k})} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right) + \left(\hat{x}_{k|k}^{(i_{1:k})} - \hat{x}_{k|k} \right) \left(\hat{x}_{k|k}^{(i_{1:k})} - \hat{x}_{k|k} \right)^T \right] \mu_{1:k}^{(i_{1:k})} \\ &= \sum_{i \in M} \left[\text{MSE} \left(\hat{x}_{k|k}^{(i)} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right) + \left(\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k} \right) \left(\hat{x}_{k|k}^{(i)} - \hat{x}_{k|k} \right)^T \right] \mu_k^{(i)}\end{aligned}\quad (15.280)$$

which holds for an estimator, $\hat{x}_{k|k}$.

Correspondingly, the **CMM-MMSE modal state estimator** at time step k is given by

$$\begin{aligned}\hat{\mathcal{M}}_{k|k} &= \mathbb{E} [\mathcal{M}_k | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}] \\ &= \sum_{i_{1:k} \in M_{1:k}} \mathbb{E} \left[\mathcal{M}_k | \vec{y}_{1:k}, \mathcal{M}_{1:k}^{(i_{1:k})} \right] \Pr \left(\mathcal{M}_{1:k}^{(i_{1:k})} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right) \\ &= \sum_{i_{1:k} \in M_{1:k}} \mathcal{M}^{(i_{1:k})} \mu_{1:k}^{(i_{1:k})} \\ &= \sum_{i \in M} \mathbb{E} \left[\mathcal{M}_k | \vec{y}_{1:k}, \mathcal{M}_k^{(i)} \right] \Pr \left(\mathcal{M}_k^{(i)} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right) \\ &= \sum_{i \in M} \hat{\mathcal{M}}_{k|k}^{(i)} \mu_k^{(i)}\end{aligned}\quad (15.281)$$

with mean square error

$$\begin{aligned}MSE(\hat{\mathcal{M}}_{k|k} | \vec{y}_{1:k}) &= \mathbb{E} \left[\left(\mathcal{M}_k - \hat{\mathcal{M}}_{k|k} \right) \left(\mathcal{M}_k - \hat{\mathcal{M}}_{k|k} \right)^T | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C} \right] \\ &= \sum_{i_{1:k} \in M_{1:k}} \left(\mathcal{M}^{(i_{1:k})} - \hat{\mathcal{M}}_{k|k} \right) \left(\mathcal{M}^{(i_{1:k})} - \hat{\mathcal{M}}_{k|k} \right)^T \mu_{1:k}^{(i_{1:k})} \\ &= \sum_{i \in M} \left(\mathcal{M}^{(i)} - \hat{\mathcal{M}}_{k|k} \right) \left(\mathcal{M}^{(i)} - \hat{\mathcal{M}}_{k|k} \right)^T \mu_k^{(i)}\end{aligned}\quad (15.282)$$

For the MAP estimators, one can note that the mixed PDF-PMF density function of the base state at time step k and mode sequence through time step k is

$$\begin{aligned} p(\vec{x}_k, \mathcal{M}_{1:k} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}) &= f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_{1:k}) p(\mathcal{M}_{1:k} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}) \\ &= \{f_{(i_{1:k})}(\vec{x}_k | \vec{y}_{1:k}) \mu_{1:k}^{(i_{1:k})}, i_{1:k} \in \mathcal{M}_{1:k}\} \\ &\neq f(\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_k^{(i)}) p(\mathcal{M}_{1:k} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}) \\ &= \{f_{(i)}(\vec{x}_k | \vec{y}_{1:k}) \mu_k^{(i)}, i \in \mathcal{M}\} \end{aligned} \quad (15.283)$$

where $f_{(i_{1:k})}(\vec{x}_k | \vec{y}_{1:k}) = f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_{1:k})$ is the PDF assuming the mode sequence is $\mathcal{M}_{1:k}^{(i_{1:k})}$.

Thus, the base state has the *a posteriori* mixed density function

$$\begin{aligned} f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}) &= \sum_{i_{1:k} \in \mathcal{M}_{1:k}} f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_{1:k}^{(i_{1:k})}) \Pr(\mathcal{M}_{1:k}^{(i_{1:k})} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}) \\ &= \sum_{i_{1:k} \in \mathcal{M}_{1:k}} f_{(i_{1:k})}(\vec{x}_k | \vec{y}_{1:k}) \mu_{1:k}^{(i_{1:k})} \\ &\neq \sum_{i \in \mathcal{M}} f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k}, \mathcal{M}_k^{(i)}) \Pr(\mathcal{M}_k^{(i)} | \vec{y}_{1:k}, \mathbf{B}, \mathbf{C}) \\ &= \sum_{i \in \mathcal{M}} f_{(i)}(\vec{x}_k | \vec{y}_{1:k}) \mu_k^{(i)} \end{aligned} \quad (15.284)$$

where the summation over the model sequences is no longer the same as the current models.

Thus, the **CMM-MAP base state estimator** at time step k is given by

$$\hat{\vec{x}}_{k|k}^{MAP} = \operatorname{argmax}_{\vec{x}_k} \sum_{i_{1:k} \in \mathcal{M}_{1:k}} f_{(i_{1:k})}(\vec{x}_k | \vec{y}_{1:k}) \mu_{1:k}^{(i_{1:k})} \quad (15.285)$$

and the **CMM-MAP mode sequence estimator** at time step k is given by

$$\hat{\mathcal{M}}_{1:k|k}^{MAP} = \operatorname{argmax}_{\mathcal{M}_{1:k}^{(i_{1:k})}} \{\mu_{1:k}^{(i_{1:k})}, i_{1:k} \in \mathcal{M}_{1:k}\} \quad (15.286)$$

It should be noted that while the entire mixed density function, $f_{(i_{1:k})}(\vec{x}_k | \vec{y}_{1:k})$, is needed to compute $\hat{\vec{x}}_{k|k}^{MAP}$, only the first two moments, $\hat{\vec{x}}_{k|k}^{(i_{1:k})}$ and $P_{k|k}^{(i_{1:k})}$, are needed to compute $\hat{\vec{x}}_{k|k}^{MMSE}$, i.e. computing $\hat{\vec{x}}_{k|k}^{MAP}$ is a maximization problem given every component density while computing $\hat{\vec{x}}_{k|k}^{MMSE}$ is an integration problem. Regardless, since the number of possible model sequences increases exponentially with time, any brute-force implementations of the above optimal CMM-MMSE or -MAP is infeasible. Thus, cooperation strategies have been proposed to overcome this difficulty in practice. These strategies are merging of “similar” model sequences, pruning “unlikely” model sequences, randomly selecting a subset of possible model sequences, or iterative strategies. The development of each of these strategies is beyond the scope of this course. However, as an introduction to this filtering concept, the interacting multi-model filter will be presented in the following subsection.

Interacting Multi-Model Filter

The **interacting multi-model (IMM) filter** uses $n_M = |\mathbb{M}|$ elemental filters in its bank and consists of four steps that are run each cycle of the filtering algorithm: model-conditioned filtering, mode probability update, estimate fusion, and model-conditioned re-initialization. The model-conditioned filtering step consists of running the n_M elemental filters, e.g. EKFs, UKFs, PFs. The mode probability update uses the current innovation distribution conditioned on the past measurements to form the model likelihoods and mode probabilities. Lastly, the overall base-state estimate and covariance can be computed using the n_M filter estimates and mode probabilities. As an example, the **interacting multi-model Kalman filter (IMM-KF)** uses Kalman filters in its elemental filter bank and can be outlined as follows.

The model-conditioned re-initialization serves as the “mixing” step of the IMM filter which re-initializes all the n_M filters based on weighted sums of the probabilities of each mode, the mode transition probabilities, and all n_M filter outputs, conditioned on the measurement history. In particular, the re-initialized state for each $i = 1, \dots, n_M$ filter is computed as

$$\begin{aligned}\bar{x}_{k-1|k-1}^{(i)} &= \mathbb{E} \left[\vec{x}_{k-1} | \vec{y}_{1:k}, \mathcal{M}_k^{(i)} \right] \\ &= \mathbb{E} \left[\mathbb{E} \left[\vec{x}_{k-1} | \vec{y}_{1:k}, \mathcal{M}_k^{(i)}, \mathcal{M}_{k-1}^{(j)} \right] | \vec{y}_{1:k}, \mathcal{M}_k^{(i)} \right] \\ &= \sum_{j \in \mathbb{M}} \hat{x}_{k|k}^{(j)} \Pr \left(\mathcal{M}_{k-1}^{(j)} | \vec{y}_{1:k}, \mathcal{M}_k^{(i)} \right)\end{aligned}\quad (15.287)$$

This dependence of the re-initialization on $\mathcal{M}_k^{(i)}$ which implies an individualized re-initialization of each filter i is the two primary benefit of IMM as opposed to other CMM filters and has been shown to be more accurate in many instances, albeit, at more computations required at each step.

The **IMM-KF mixing step** computes the following for $i = 1, \dots, n_M$. The i^{th} predicted mode probability is given by

$$\mu_{k|k-1}^{(i)} = \Pr \left(\mathcal{M}_k^{(i)} | \vec{y}_{1:k} \right) = \sum_{j \in \mathbb{M}} \pi_{ji} \mu_{k-1|k-1}^{(j)} \quad (15.288)$$

The i^{th} mixing weight is given by

$$\mu_{k-1}^{j|i} = \Pr \left(\mathcal{M}_k^{(i)} | \vec{y}_{1:k} \right) = \pi_{ji} \frac{\mu_{k-1|k-1}^{(j)}}{\mu_{k|k-1}^{(i)}} \quad (15.289)$$

The i^{th} mixing estimate is given by

$$\bar{x}_{k-1|k-1}^{(i)} = \mathbb{E} \left[\vec{x}_{k-1} | \mathcal{M}_k^{(i)}, \vec{y}_{1:k} \right] = \sum_{j \in \mathbb{M}} \mu_{k-1}^{j|i} \hat{x}_{k-1|k-1}^{(j)} \quad (15.290)$$

The i^{th} mixing covariance is given by

$$\bar{P}_{k-1|k-1}^{(i)} = \sum_{j \in \mathbb{M}} \mu_{k-1}^{j|i} \left[P_{k-1|k-1}^{(j)} + \left(\bar{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)} \right) \left(\bar{x}_{k-1|k-1}^{(i)} - \hat{x}_{k-1|k-1}^{(j)} \right)^T \right] \quad (15.291)$$

The **IMM-KF filtering step** computes the following for $i = 1, \dots, n_M$. The i^{th} predicted state is given by

$$\hat{x}_{k|k-1}^{(i)} = F_{k-1}^{(i)} \bar{x}_{k-1|k-1}^{(i)} + G_{k-1}^{(i)} \bar{w}_{k-1}^{(i)} \quad (15.292)$$

The i^{th} predicted covariance is given by

$$P_{k|k-1}^{(i)} = F_{k-1}^{(i)} \bar{P}_{k-1|k-1}^{(i)} \left(F_{k-1}^{(i)} \right)^T + G_{k-1}^{(i)} Q_{k-1}^{(i)} \left(G_{k-1}^{(i)} \right)^T \quad (15.293)$$

The i^{th} innovation is given by

$$\tilde{y}_k^{(i)} = \vec{y}_k - H_k^{(i)} \hat{x}_{k|k-1}^{(i)} - \bar{v}_k^{(i)} \quad (15.294)$$

The i^{th} innovation covariance is given by

$$S_k^{(i)} = H_k^{(i)} P_{k|k-1}^{(i)} \left(H_k^{(i)} \right)^T + R_k^{(i)} \quad (15.295)$$

The i^{th} Kalman filter gain is given by

$$K_k^{(i)} = P_{k|k-1}^{(i)} \left(H_k^{(i)} \right)^T \left(S_k^{(i)} \right)^{-1} \quad (15.296)$$

The i^{th} corrected state is given by

$$\hat{x}_{k|k}^{(i)} = \hat{x}_{k|k-1}^{(i)} + K_k^{(i)} \tilde{y}_k^{(i)} \quad (15.297)$$

The i^{th} corrected covariance is given by

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} S_k^{(i)} \left(K_k^{(i)} \right)^T \quad (15.298)$$

The **IMM-KF mode probability update step** computes the following for $i = 1, \dots, n_M$. The i^{th} model likelihood is given by

$$\mathcal{L}_k^{(i)} = f_{\vec{Y}} \left(\tilde{y}_k^{(i)} | \mathcal{M}_k^{(i)}, \vec{y}_{1:k} \right) = f_{\mathcal{N}} \left(\tilde{y}_k^{(i)}; 0, S_k^{(i)} \right) \quad (15.299)$$

where the multivariate Gaussian model is assumed in this case. The i^{th} mode probability is given by

$$\mu_{k|k}^{(i)} = \frac{\mu_{k|k-1}^{(i)} \mathcal{L}_k^{(i)}}{\sum_{j \in \mathcal{M}} \mu_{k|k-1}^{(j)} \mathcal{L}_k^{(j)}} \quad (15.300)$$

The **IMM-KF estimate fusion step** computes the overall base-state estimate as

$$\hat{x}_{k|k} = \sum_{i \in \mathcal{M}} \mu_{k|k}^{(i)} \hat{x}_{k|k}^{(i)} \quad (15.301)$$

and the overall base-state covariance as

$$P_{k|k} = \sum_{i \in \mathcal{M}} \mu_{k|k}^{(i)} \left[P_{k|k}^{(i)} + \left(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)} \right) \left(\hat{x}_{k|k} - \hat{x}_{k|k}^{(i)} \right)^T \right] \quad (15.302)$$

which is the approximate MMSE and MAP for the Gaussian mixture assumption for this filter.

By inspection, one can see that the elemental filters could easily be extended to any Bayes-based filter, e.g. the EKF. It is also important to note that one must be able to model the modal transition probabilities, π_{ji} , appropriately for the IMM to work well, i.e. either constant if S_k is assumed to be a Markov process, or a sojourn-time dependent model if S_k is assumed to be a semi-Markov process.

References

For more information, please refer to the following

- Bar-Shalom, Y., Daum, F., and Huang, J., “The Probabilistic Data Association Filter,” in *IEEE Control Systems Magazine*, Vol. 29, Issue 6, 2009, pp. 82-100
- Simon, D., “10.2 Multiple-Model Estimation,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 301-303

Introductory Nonlinear Bayesian State Estimation Theory

16.1 Introduction to Nonlinear Bayesian State Estimation

Recall the Bayes filter is the optimal recursive solution to the general state estimation problem and can be written as prediction step for the prior PDF

$$f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k-1}) = \int_{-\infty}^{\infty} f_{\vec{X}}(\vec{x}_k | \vec{x}_{k-1}) f_{\vec{X}}(\vec{x}_{k-1} | \vec{y}_{1:k-1}) d\vec{x}_{k-1} \quad (16.1)$$

and correction step for the posterior PDF

$$f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k}) = \frac{f_{\vec{Y}}(\vec{y}_k | \vec{x}_k) f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k-1})}{\int_{-\infty}^{\infty} f_{\vec{Y}}(\vec{y}_k | \vec{x}_k) f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k-1}) d\vec{x}_k} \quad (16.2)$$

which usually only provides an exact closed-form solutions for linear-Gaussian systems, i.e., the discrete-time, continuous-time, and hybrid-time Kalman filters, and the Benes filter. Beyond these case, exact closed-form recursive solutions do not exist and one must make approximations to the optimal Bayes filter for nonlinear, non-Gaussian stochastic state-space models which are discussed in this chapter. Typically these filters are either **assumed density filters (ADF)** or **Monte-Carlo filters**.

Discrete-Time Nonlinear Kalman Filtering

Motivated by the optimality of the discrete-time Kalman filter, one of the most common approximations is to approximate the prior and posterior distributions of the optimal Bayes filter as multivariate Gaussians. This approach is known as discrete-time **nonlinear Kalman filtering**, also known as **Gaussian filtering**, where the state mean/estimate, \hat{x}_k , and the state covariance, P_k , are recursively updated as hyper-parameters representing the multivariate Gaussian approximation as opposed to calculating an exact posterior conditional PDF, $f_{\vec{X}}(\vec{x}_k | \vec{y}_{1:k})$, which may not be possible. In this context, consider the discrete-time stochastic state-

space model as

$$\begin{aligned}\vec{x}_k &= f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}) \\ \vec{y}_k &= h_k(\vec{x}_k, \vec{v}_k)\end{aligned}\quad (16.3)$$

By definition of the mean and covariance, a discrete-time nonlinear Kalman filter has a prediction step for the prior mean given by

$$\hat{\vec{x}}_{k|k-1} = \mathbb{E}[f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1})] \quad (16.4)$$

the discrete-time prior covariance given by

$$P_{k|k-1} = \mathbb{E}[(\vec{x}_k - \hat{\vec{x}}_{k|k-1})(\vec{x}_k - \hat{\vec{x}}_{k|k-1})^T] \quad (16.5)$$

and the predicted measurement given by

$$\hat{\vec{y}}_k = \mathbb{E}[h_k(\vec{x}_k, \vec{v}_k)] \quad (16.6)$$

as well as a correction step for the measurement covariance given by

$$P_{y_k} = S_k = \mathbb{E}[(\vec{y}_k - \hat{\vec{y}}_k)(\vec{y}_k - \hat{\vec{y}}_k)^T] \quad (16.7)$$

the state-measurement cross-covariance given by

$$P_{x_k y_k} = \mathbb{E}[(\vec{x}_k - \hat{\vec{x}}_{k|k-1})(\vec{y}_k - \hat{\vec{y}}_k)^T] \quad (16.8)$$

the optimal gain given by

$$K_k = P_{x_k y_k} P_{y_k}^{-1} = P_{x_k y_k} S_k^{-1} \quad (16.9)$$

the posterior mean given by

$$\hat{\vec{x}}_{k|k} = \hat{\vec{x}}_{k|k-1} + K_k [\vec{y}_k - \hat{\vec{y}}_k] \quad (16.10)$$

and the posterior covariance given by

$$P_{k|k} = P_{k|k-1} - K_k P_{y_k} K_k^T \quad (16.11)$$

The conceptually most straightforward nonlinear Kalman filter is the extended Kalman filter (EKF) which approximates the expectation integrals for the nonlinear Kalman filter using a first-order Jacobian linearization of the nonlinear system about the current state estimate. This approach also allows one to easily apply this linearization to the continuous- and hybrid-time stochastic state-space models. However, the EKF disregards the “probabilistic spread” of the state vector as the Jacobian linearization is defined about a *single* point. Thus, this approximation can produce large errors of the approximated prior and posterior mean and covariance relative to the true prior and posterior mean and covariance which leads to poor performance and even divergence.

Discrete-Time Extended Kalman Filter

Consider the Jacobian linearization of the process equation about the posterior mean, $\hat{\vec{x}}_{k-1|k-1}$, and the mean of the process noise, $\vec{0}$, i.e.,

$$\begin{aligned}\vec{x}_k &= f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}) \approx f_{k-1}(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, \vec{0}) \\ &\quad + \frac{\partial f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1})}{\partial \vec{x}_{k-1}} \Big|_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} (\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) \\ &\quad + \frac{\partial f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1})}{\partial \vec{w}_{k-1}} \Big|_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} (\vec{w}_{k-1} - \vec{0})\end{aligned}\quad (16.12)$$

Then, defining the linearized state matrix as

$$F_{k-1} = \frac{\partial f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1})}{\partial \vec{x}_{k-1}} \Big|_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \quad (16.13)$$

and the linearized process noise gain matrix

$$L_{k-1} = \frac{\partial f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1})}{\partial \vec{w}_{k-1}} \Big|_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \quad (16.14)$$

one has

$$\vec{x}_k \approx f_{k-1}(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, 0) + F_{k-1}(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) + L_{k-1} \vec{w}_{k-1} \quad (16.15)$$

By definition, the prior state mean is given by

$$\hat{\vec{x}}_{k|k-1} = \mathbb{E}[f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1})] \quad (16.16)$$

Substituting by the approximation, one has

$$\hat{\vec{x}}_{k|k-1} \approx \mathbb{E}\left[f_{k-1}(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, \vec{0}) + F_{k-1}(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) + L_{k-1} \vec{w}_{k-1}\right] \quad (16.17)$$

which by definition that $\mathbb{E}(\hat{\vec{x}}_{k|k-1}) = \vec{x}_k$ and zero-mean noise, one has

$$\hat{\vec{x}}_{k|k-1} \approx f_{k-1}(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, \vec{0}) \quad (16.18)$$

By definition, the prior state covariance is given by

$$P_{k|k-1} = \mathbb{E}[(\vec{x}_k - \hat{\vec{x}}_{k|k-1})(\vec{x}_k - \hat{\vec{x}}_{k|k-1})^T] \quad (16.19)$$

Substituting by the approximation for \vec{x}_k and the prior state mean, one has

$$P_{k|k-1} = \mathbb{E}\left[(F_{k-1}(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) + L_{k-1} \vec{w}_{k-1})(F_{k-1}(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) + L_{k-1} \vec{w}_{k-1})^T\right] \quad (16.20)$$

or

$$\begin{aligned}P_{k|k-1} &= F_{k-1} \mathbb{E}\left[(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})^T\right] F_{k-1}^T + L_{k-1} \mathbb{E}\left[\vec{w}_{k-1} \vec{w}_{k-1}^T\right] L_{k-1}^T \\ &\quad F_{k-1} \mathbb{E}\left[(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) \vec{w}_{k-1}^T\right] L_{k-1}^T + L_{k-1} \mathbb{E}\left[\vec{w}_{k-1} (\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})^T\right] F_{k-1}^T\end{aligned}\quad (16.21)$$

For uncorrelated state and process noise, one has

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \quad (16.22)$$

Consider the Jacobian linearization of the measurement equation about the prior mean, $\hat{\vec{x}}_{k|k-1}$, and the mean of the measurement noise, $\vec{0}$, i.e.,

$$\vec{y}_k = h_k(\vec{x}_k, \vec{v}_k) \approx h_k(\hat{\vec{x}}_{k|k-1}, \vec{0}) + \frac{\partial h_k(\vec{x}_k, \vec{v}_k)}{\partial \vec{x}_k} \Big|_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} (\vec{x}_k - \hat{\vec{x}}_{k|k-1}) + \frac{\partial h_k(\vec{x}_k, \vec{v}_k)}{\partial \vec{v}_k} \Big|_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} (\vec{v}_k - \vec{0}) \quad (16.23)$$

Then, defining the linearized output matrix as

$$H_k = \frac{\partial h_k(\vec{x}_k, \vec{v}_k)}{\partial \vec{x}_k} \Big|_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} \quad (16.24)$$

and the linearized measurement noise gain matrix

$$M_k = \frac{\partial h_k(\vec{x}_k, \vec{v}_k)}{\partial \vec{v}_k} \Big|_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} \quad (16.25)$$

one has

$$\vec{y}_k \approx h_k(\hat{\vec{x}}_{k|k-1}, \vec{0}) + H_k(\vec{x}_k - \hat{\vec{x}}_{k|k-1}) + M_k \vec{v}_k \quad (16.26)$$

By definition, the predicted measurement is given by

$$\hat{\vec{y}}_k = \mathbb{E}[h_k(\vec{x}_k, \vec{v}_k)] \quad (16.27)$$

Substituting by the approximation, one has

$$\hat{\vec{y}}_k \approx \mathbb{E}\left[h_k(\hat{\vec{x}}_{k|k-1}, \vec{0}) + H_k(\vec{x}_k - \hat{\vec{x}}_{k|k-1}) + M_k \vec{v}_k\right] \quad (16.28)$$

which by definition that $\mathbb{E}(\hat{\vec{x}}_{k|k-1}) = \vec{x}_k$ and zero-mean noise, one has

$$\hat{\vec{y}}_k \approx h_k(\hat{\vec{x}}_{k|k-1}, \vec{0}) \quad (16.29)$$

By definition, the measurement covariance is given by

$$S_k = \mathbb{E}[(\vec{y}_k - \hat{\vec{y}}_k)(\vec{y}_k - \hat{\vec{y}}_k)^T] \quad (16.30)$$

Substituting by the approximation for \vec{y}_k and the predicted measurement, one has

$$S_k = \mathbb{E}\left[(H_k(\vec{x}_k - \hat{\vec{x}}_{k|k-1}) + M_k \vec{v}_k)(H_k(\vec{x}_k - \hat{\vec{x}}_{k|k-1}) + M_k \vec{v}_k)^T\right] \quad (16.31)$$

or

$$\begin{aligned} S_k = & H_k \mathbb{E}\left[(\vec{x}_k - \hat{\vec{x}}_{k|k-1})(\vec{x}_k - \hat{\vec{x}}_{k|k-1})^T\right] H_k^T + M_k \mathbb{E}\left[\vec{v}_k \vec{v}_k^T\right] M_k^T \\ & H_k \mathbb{E}\left[(\vec{x}_k - \hat{\vec{x}}_{k|k-1}) \vec{v}_k^T\right] M_k^T + M_k \mathbb{E}\left[\vec{v}_k (\vec{x}_k - \hat{\vec{x}}_{k|k-1})^T\right] H_k^T \end{aligned} \quad (16.32)$$

For uncorrelated state and measurement noise, one has

$$S_k = H_k P_{k|k-1} H_k^T + M_k R_{k-1} M_k^T \quad (16.33)$$

By definition, the state-measurement cross-covariance given by

$$P_{x_k y_k} = \mathbb{E} [(\vec{x}_k - \hat{\vec{x}}_{k|k-1})(\vec{y}_k - \hat{\vec{y}}_k)^T] \quad (16.34)$$

Substituting by the approximation for \vec{y}_k and the predicted measurement, one has

$$P_{x_k y_k} = \mathbb{E} [(F_{k-1}(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) + L_{k-1} \vec{w}_{k-1})(H_k(\vec{x}_k - \vec{x}_{k|k-1}) + M_k \vec{v}_k)^T] \quad (16.35)$$

$$\begin{aligned} P_{x_k y_k} = & \mathbb{E} [F_{k-1}(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})(\vec{x}_k - \vec{x}_{k|k-1})] H_k^T + L_{k-1} \vec{w}_{k-1} \vec{v}_k^T M_k^T \\ & + F_{k-1} \mathbb{E} [(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1}) \vec{v}_k^T] M_k^T + L_{k-1} \mathbb{E} [\vec{w}_{k-1} (\vec{x}_k - \vec{x}_{k|k-1})^T] H_k^T \end{aligned} \quad (16.36)$$

For uncorrelated state, process noise, and measurement noise, one has

$$P_{x_k y_k} = F_{k-1} \mathbb{E} [F_{k-1}(\vec{x}_{k-1} - \hat{\vec{x}}_{k-1|k-1})(\vec{x}_{k-1} - \vec{x}_{k-1|k-1})^T F_{k-1}^T] H_k^T \quad (16.37)$$

or

$$P_{x_k y_k} = P_{k|k-1} H_k^T \quad (16.38)$$

In summary, the discrete-time **EKF initialization step** sets the state estimate and state covariance at $k = 0$ as

$$\begin{aligned} \hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 = \mathbb{E} [\vec{x}_0] \\ P_{0|0} &= P_0 = \mathbb{E} [(\vec{x}_0 - \hat{\vec{x}}_{0|0})(\vec{x}_0 - \hat{\vec{x}}_{0|0})^T] \end{aligned} \quad (16.39)$$

The discrete-time **EKF prediction step** predicts the state estimate, the state covariance, and the output using the input, process model, and Jacobians as

$$\begin{aligned} \hat{\vec{x}}_{k|k-1} &= f(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, \vec{0}) \\ \hat{\vec{y}}_k &= h(\hat{\vec{x}}_{k|k-1}, \vec{0}) \\ F_{k-1} &= \left[\frac{\partial f}{\partial \vec{x}} \right]_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \\ L_{k-1} &= \left[\frac{\partial f}{\partial \vec{w}} \right]_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \\ P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \end{aligned} \quad (16.40)$$

The discrete-time **EKF correction step** updates the state estimate and the state covariance using the measurement model and Jacobians as

$$\begin{aligned}
 \tilde{y}_k &= \vec{y}_k - \hat{\vec{y}}_k \\
 H_k &= \left[\frac{\partial h}{\partial \vec{x}} \right]_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} \\
 M_k &= \left[\frac{\partial h}{\partial \vec{v}} \right]_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} \\
 S_k &= H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \\
 K_k &= P_{k|k-1} H_k^T S_k^{-1} \\
 \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k \tilde{y}_k \\
 P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T
 \end{aligned} \tag{16.41}$$

Continuous-Time Extended Kalman Filter

Consider the continuous-time stochastic time-invariant state-space model about some nominal trajectory

$$\begin{aligned}
 \dot{\vec{x}}_0 &= f(\vec{x}_0, \vec{u}_0, \vec{w}_0, t) \\
 \vec{y}_0 &= h(\vec{x}_0, \vec{v}_0, t)
 \end{aligned} \tag{16.42}$$

Next, define the linearized state matrix as

$$A = \left[\frac{\partial f}{\partial \vec{x}} \right]_{\vec{x}=\vec{x}_0, \vec{w}=\vec{w}_0} \tag{16.43}$$

the linearized process noise gain matrix

$$L = \left[\frac{\partial f}{\partial \vec{w}} \right]_{\vec{x}=\vec{x}_0, \vec{w}=\vec{w}_0} \tag{16.44}$$

the linearized output matrix as

$$C = \left[\frac{\partial h}{\partial \vec{x}} \right]_{\vec{x}=\vec{x}_0, \vec{v}=\vec{v}_0} \tag{16.45}$$

and the linearized measurement noise gain matrix

$$M = \left[\frac{\partial h}{\partial \vec{v}} \right]_{\vec{x}=\vec{x}_0, \vec{v}=\vec{v}_0} \tag{16.46}$$

Then, the Jacobian linearization can be written as

$$\begin{aligned}
 \dot{\vec{x}} &= f(\vec{x}, \vec{u}, \vec{w}) = f(\vec{x}_0, \vec{u}_0, \vec{w}_0) + A(\vec{x} - \vec{x}_0) + B(\vec{u} - \vec{u}_0) + L(\vec{w} - \vec{w}_0) \\
 \vec{y} &= h(\vec{x}, \vec{v}) = h(\vec{x}_0, \vec{v}_0) + C(\vec{x} - \vec{x}_0) + M(\vec{v} - \vec{v}_0)
 \end{aligned} \tag{16.47}$$

or using perturbation notation and the definitions of $\dot{\vec{x}}_0$ and \vec{y}_0 , one has

$$\begin{aligned}\Delta \dot{\vec{x}} &= A\Delta \vec{x} + B\Delta \vec{u} + L\Delta \vec{w} \\ \Delta \vec{y} &= C\Delta \vec{x} + M\Delta \vec{v}\end{aligned}\quad (16.48)$$

and assuming $\vec{u}(t)$ is perfectly known, i.e., $\Delta u(t) = 0$, and $\vec{w}_0 = \vec{v}_0 = \vec{0}$, one can write

$$\begin{aligned}\Delta \dot{\vec{x}} &= A\Delta \vec{x} + L\vec{w} \\ \Delta \vec{y} &= C\Delta \vec{x} + M\vec{v}\end{aligned}\quad (16.49)$$

where \vec{w} and \vec{v} have covariances, Q and R , respectively.

These equations form a linear dynamical system of the state perturbation, $\Delta \hat{\vec{x}}$, which can be estimated using the continuous-time Kalman filter as

$$\begin{aligned}\dot{P} &= AP + PA^T - PC^T(MRM^T)^{-1}CP + LQL^T \\ K &= PC^T(MRM^T)^{-1} \\ \Delta \dot{\hat{\vec{x}}} &= A\Delta \hat{\vec{x}} + K(\Delta \vec{y} - C\Delta \hat{\vec{x}})\end{aligned}\quad (16.50)$$

and added to the nominal estimator to obtain

$$\hat{\vec{x}} = \vec{x}_0 + \Delta \hat{\vec{x}} \quad (16.51)$$

Furthermore, note that one can write

$$\dot{\vec{x}}_0 + \Delta \dot{\hat{\vec{x}}} = f(\vec{x}_0, \vec{u}_0, \vec{0}, t) + A\Delta \hat{\vec{x}} + K(\vec{y} - \vec{y}_0 - C\Delta \hat{\vec{x}}) \quad (16.52)$$

If one chooses $\vec{x}_0(t) = \hat{\vec{x}}(t)$ so that $\Delta \hat{\vec{x}}(t) = 0$ and $\Delta \dot{\hat{\vec{x}}}(t) = 0$, i.e., the linearization trajectory is equivalent to the linearized Kalman filter state estimate, $\hat{\vec{x}}(t)$, which also infers the nominal measurement as

$$\vec{y}_0 = h(\hat{\vec{x}}, \vec{0}, t) \quad (16.53)$$

and assume that one has some *prior* estimate of the state $\hat{\vec{x}}_0$ with some expected uncertainty P_0 where

$$\mathbb{E}[\vec{x}] = \hat{\vec{x}}_0 \quad (16.54)$$

and

$$\mathbb{E}[(\vec{x} - \hat{\vec{x}}_0)(\vec{x} - \hat{\vec{x}}_0)^T] = P_0 \quad (16.55)$$

$$\begin{aligned}\hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 = \\ P_{0|0} &= P_0\end{aligned}\quad (16.56)$$

Then, one obtains the **continuous-time extended Kalman filter (EKF)**, also known as the **extended Kalman-Bucy filter (EKBF)**, has an initialization step as

$$\begin{aligned}\hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 = \mathbb{E} [\vec{x}_0] \\ P_{0|0} &= P_0 = \mathbb{E} [(\vec{x}_0 - \hat{\vec{x}}_{0|0})(\vec{x}_0 - \hat{\vec{x}}_{0|0})^T] \\ A &= \left[\frac{\partial f}{\partial \vec{x}} \right]_{\vec{x}=\hat{\vec{x}}, \vec{w}=\vec{0}} \\ L &= \left[\frac{\partial f}{\partial \vec{w}} \right]_{\vec{x}=\hat{\vec{x}}, \vec{w}=\vec{0}} \\ C &= \left[\frac{\partial h}{\partial \vec{x}} \right]_{\vec{x}=\hat{\vec{x}}, \vec{v}_k=\vec{0}} \\ M &= \left[\frac{\partial h}{\partial \vec{v}} \right]_{\vec{x}=\hat{\vec{x}}, \vec{v}_k=\vec{0}}\end{aligned}\tag{16.57}$$

and differential equations as

$$\begin{aligned}\dot{P} &= AP + PA^T - PC^T(MRM^T)^{-1}CP + LQL^T \\ K &= PC^T(MRM^T)^{-1} \\ \dot{\hat{\vec{x}}} &= f(\hat{\vec{x}}, \vec{u}, \vec{0}, t) + K(\vec{y} - h(\hat{\vec{x}}, \vec{0}, t))\end{aligned}\tag{16.58}$$

Hybrid-Time Extended Kalman Filter

Lastly, it should be noted that often in practice, one has continuous-time dynamics and discrete-time measurements, which is known as a **hybrid-time stochastic state-space model**

$$\begin{aligned}\dot{\vec{x}}(t) &= f(\vec{x}(t), \vec{u}(t), \vec{w}(t)) \\ \vec{y}_k &= h(\vec{x}_k, \vec{v}_k)\end{aligned}\tag{16.59}$$

for which one can form the **hybrid-time EKF** by utilizing the nonlinear differential equation and the Riccati differential equation for the prediction step and the discrete-time EKF equations for the correction step. The hybrid-time EKF can be considered as an alternative to discretizing $f(\vec{x}(t), \vec{u}(t), \vec{w}(t))$ using and implementing a discrete-time EKF. This type of discretization may or may not be tractable for general nonlinear dynamical systems. Both discrete-time and hybrid-time use integrals for propagation or discretization which may require numerical methods.

In short, the hybrid-time **EKF initialization step** sets the state estimate and state covariance at $k = 0$ as

$$\begin{aligned}\hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 = \mathbb{E} [\vec{x}_0] \\ P_{0|0} &= P_0 = \mathbb{E} [(\vec{x}_0 - \hat{\vec{x}}_{0|0})(\vec{x}_0 - \hat{\vec{x}}_{0|0})^T]\end{aligned}\tag{16.60}$$

The hybrid-time **EKF prediction step** predicts the state estimate, the state covariance, and the output using the input, process model, and Jacobians as

$$\begin{aligned}
 \hat{\vec{x}}_{k-1}(t) &= \int_{t(k-1)}^{t(k)} f(\hat{\vec{x}}, \vec{u}, 0) \text{ with } \hat{\vec{x}}_{k-1}(t(k-1)) = \hat{\vec{x}}_{k-1|k-1} \\
 \hat{\vec{x}}_{k|k-1} &= \hat{\vec{x}}_{k-1}(t(k)) \\
 \hat{\vec{y}}_k &= h(\hat{\vec{x}}_{k|k-1}, \vec{0}) \\
 A_{k-1}(t) &= \left[\frac{\partial f}{\partial \vec{x}} \right]_{\vec{x}=\hat{\vec{x}}_{k-1}(t), \vec{w}_{k-1}=\vec{0}} \\
 L_{k-1}(t) &= \left[\frac{\partial f}{\partial \vec{w}} \right]_{\vec{x}=\hat{\vec{x}}_{k-1}(t), \vec{w}_{k-1}=\vec{0}} \\
 P_{k|k-1} &= P_{k-1|k-1} + \int_{t(k-1)}^{t(k)} A_{k-1}P + PA_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T
 \end{aligned} \tag{16.61}$$

The hybrid-time **EKF correction step** updates the state estimator and the state covariance using the measurement model and Jacobians as

$$\begin{aligned}
 \tilde{\vec{y}}_k &= \vec{y}_k - \hat{\vec{y}}_k \\
 H_k &= \left[\frac{\partial h}{\partial \vec{x}} \right]_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} \\
 M_k &= \left[\frac{\partial h}{\partial \vec{v}} \right]_{\vec{x}_k=\hat{\vec{x}}_{k|k-1}, \vec{v}_k=\vec{0}} \\
 S_k &= H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \\
 K_k &= P_{k|k-1} H_k^T S_k^{-1} \\
 \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k \tilde{\vec{y}}_k \\
 P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T
 \end{aligned} \tag{16.62}$$

References

For more information, please refer to the following

- Sarkka, S., “5.1 Taylor series expansions,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 64-69
- Sarkka, S., “5.2 Extended Kalman filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 69-75
- Simon, D., “13.1 The linearized Kalman filter,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 397-399
- Simon, D., “13.2 The extended Kalman filter,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 407-410

16.2 Alternative Extended Kalman Filtering and Smoothing

Iterative Extended Kalman Filter

At each correction step for the discrete- and hybrid-time EKFs, the Jacobian matrices depend on the prior state estimate, $\vec{x}_{k|k-1}$, which, in turn, affects the Kalman gain computation, the posterior state estimate, and the posterior state covariance. An alternative EKF approach is the **iterative extended Kalman filter (IEKF)** which iterates across some fixed number of iterations, i.e., for $i = 0, 1, \dots, N$, for the Jacobian linearizations of the measurement matrix, $H_{k,i}$, the measurement noise gain matrix, $M_{k,i}$, the innovation covariance, $S_{k,i}$, and the subsequent Kalman gain, $K_{k,i}$, based on iterative state estimates in the correction step.

In this case, one must also update innovation using a first-order Taylor series approximation evaluated at the iterated state estimate, $\hat{\vec{x}}_{k,i}$, i.e.,

$$\tilde{y}_k = \vec{y}_k - h(\vec{x}_{k|k-1}, \vec{v}_k) \approx h(\vec{x}_{k,i}, \vec{v}_k) + H_{k,i}(\hat{\vec{x}}_{k|k-1} - \hat{\vec{x}}_{k,i}) \quad (16.63)$$

In summary, the discrete-time **IEKF initialization step** sets the state estimate and state covariance at $k = 0$ as

$$\begin{aligned} \hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 = \mathbb{E}[\vec{x}_0] \\ P_{0|0} &= P_0 = \mathbb{E}[(\vec{x}_0 - \hat{\vec{x}}_{0|0})(\vec{x}_0 - \hat{\vec{x}}_{0|0})^T] \end{aligned} \quad (16.64)$$

The discrete-time **IEKF prediction step** predicts the state estimate, the state covariance, and the output using the input, process model, and Jacobians as

$$\begin{aligned} \hat{\vec{x}}_{k|k-1} &= f_{k-1}(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, \vec{0}) \\ F_{k-1} &= \left[\frac{\partial f_{k-1}}{\partial \vec{x}} \right]_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \\ L_{k-1} &= \left[\frac{\partial f_{k-1}}{\partial \vec{w}} \right]_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \\ P_{k|k-1} &= F_{k-1}P_{k-1|k-1}F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T \end{aligned} \quad (16.65)$$

which could alternatively be cast as a hybrid-time IEKF prediction step.

$$\begin{aligned} \hat{\vec{x}}_{k-1}(t) &= \int_{t(k-1)}^{t(k)} f_{k-1}(\hat{\vec{x}}, \vec{u}, 0) \text{ with } \hat{\vec{x}}_{k-1}(t(k-1)) = \hat{\vec{x}}_{k-1|k-1} \\ \hat{\vec{x}}_{k|k-1} &= \hat{\vec{x}}_{k-1}(t(k)) \\ A_{k-1}(t) &= \left[\frac{\partial f_{k-1}}{\partial \vec{x}} \right]_{\vec{x}=\hat{\vec{x}}_{k-1}(t), \vec{w}=\vec{0}} \\ L_{k-1}(t) &= \left[\frac{\partial f_{k-1}}{\partial \vec{w}} \right]_{\vec{x}=\hat{\vec{x}}_{k-1}(t), \vec{w}=\vec{0}} \\ P_{k|k-1} &= P_{k-1|k-1} + \int_{t(k-1)}^{t(k)} A_{k-1}P + PA_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T \end{aligned} \quad (16.66)$$

The discrete- and hybrid-time **IEKF correction step** iteratively updates the state estimate and the state covariance using the measurement model and Jacobians for $i = 0, 1, \dots, N$

$$\begin{aligned} H_{k,i} &= \left[\frac{\partial h_k}{\partial \vec{x}} \right]_{\vec{x}_k=\hat{\vec{x}}_{k,i}, \vec{v}_k=\vec{0}} \\ M_{k,i} &= \left[\frac{\partial h_k}{\partial \vec{v}} \right]_{\vec{x}_k=\hat{\vec{x}}_{k,i}, \vec{v}_k=\vec{0}} \\ S_{k,i} &= H_{k,i} P_{k|k-1} H_{k,i}^T + M_{k,i} R_k M_{k,i}^T \\ K_{k,i} &= P_{k|k-1} H_{k,i}^T S_{k,i}^{-1} \\ \tilde{y}_{k,i} &= \vec{y}_k - h_k(\hat{\vec{x}}_{k,i}, \vec{0}) - H_{k,i}(\hat{\vec{x}}_{k|k-1} - \hat{\vec{x}}_{k,i}) \\ \hat{\vec{x}}_{k,i+1} &= \hat{\vec{x}}_{k|k-1} + K_{k,i} \tilde{y}_{k,i} \\ P_{k,i+1} &= P_{k|k-1} - K_{k,i} S_{k,i} K_{k,i}^T \end{aligned} \quad (16.67)$$

where

$$\begin{aligned} \hat{\vec{x}}_{k,0} &= \hat{\vec{x}}_{k|k-1} \\ P_{k,0} &= P_{k|k-1} \end{aligned} \quad (16.68)$$

and

$$\begin{aligned} \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k,N+1} \\ P_{k|k} &= P_{k,N+1} \end{aligned} \quad (16.69)$$

Innovation-Saturated Extended Kalman Filter

In practice, a sensor may experience large outliers, \vec{d}_k , which corrupt the nominal measurement model used in the filter, i.e., one can model the stochastic state-space system as

$$\begin{aligned} \vec{x}_k &= f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}) \\ \vec{y}_k &= h_k(\vec{x}_k, \vec{v}_k) + \vec{d}_k \end{aligned} \quad (16.70)$$

where \vec{d}_k are spontaneous outliers which are difficult to detect or characterize which hinder the use of an augmented state. One simple method to detecting and rejecting outliers is to compute if the innovation for each element $i = 1, \dots, n_y$ is within its 99% confidence bounds, i.e., one rejects the i^{th} measurement if

$$\vec{y}_{i,k} - h_{i,k}(\hat{\vec{x}}_{k|k-1}, \vec{0}) > 3\sqrt{[S_k]_{i,i}} \quad (16.71)$$

However, this simple method may not provide suitable performance to the EKF.

Another approach to mitigate the effects of these “hybrid-measurements” is through a saturation operation on the effect of the innovation on the correction step due to potential outliers. This idea leads to the **innovation-saturation extended Kalman filter** which uses a **saturation function** for each innovation $i = 1, \dots, n_y$ as

$$\text{sat}_{\sqrt{\sigma_i}}(\tilde{y}_i) = \max\{-\sqrt{\sigma_i}, \min\{\sqrt{\sigma_i}, \tilde{y}_i\}\} \quad (16.72)$$

where $\sigma_i > 0$ is the i^{th} **saturation bound** and the range $[-\sqrt{\sigma_i}, \sqrt{\sigma_i}]$ defines the expected range of the innovation. With this function, one can dynamically adapt the i^{th} saturation bound with the dynamics model

$$\begin{aligned}\sigma_{i,k} &= \lambda_{1,i}\sigma_{i,k} + \gamma_{1,i}\epsilon_{i,k} \exp(-\epsilon_{i,k}) \\ \epsilon_{i,k} &= \lambda_{2,i}\epsilon_{i,k} + \gamma_{2,i}(\vec{y}_{i,k} - h_{i,k}(\hat{\vec{x}}_{k|k-1}, \vec{0}))^2\end{aligned}\quad (16.73)$$

with $0 < \lambda_{1,i} < 1$, $0 < \lambda_{2,i} < 1$, $\gamma_{1,i} > 0$, and $\gamma_{2,i} > 0$. These recursions serve to dynamically change $\sigma_{i,k}$ to enable adaptation of the saturation bounds.

This formulation specifically involves a double-layer structure. The lower layer for $\epsilon_{i,k}$ tracks the changes in the innovation signal, maintaining an appropriate level when the innovation is within normality, but becoming large when the innovation is altered by outliers. The upper layer for $\sigma_{i,k}$ will be driven to increase for relatively small $\epsilon_{i,k}$ when the innovation is within normality, but it will quickly reduce $\sigma_{i,k}$ when $\epsilon_{i,k}$ is large due to outliers. This adaptive approach for $\sigma_{i,k}$ is intended to achieve a discernment between an outlier and a normal measurement, filtering innovations if corrupted by outliers and passing normal innovations to correct the prediction.

Thus, the discrete-time **IS-EKF initialization step** sets the state estimate and state covariance at $k = 0$ as

$$\begin{aligned}\hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 = \mathbb{E}[\vec{x}_0] \\ P_{0|0} &= P_0 = \mathbb{E}[(\vec{x}_0 - \hat{\vec{x}}_{0|0})(\vec{x}_0 - \hat{\vec{x}}_{0|0})^T] \\ \sigma_{i,1} &> 0 \\ \epsilon_{i,1} &> 0\end{aligned}\quad (16.74)$$

The discrete-time **IS-EKF prediction step** predicts the state estimate, the state covariance, and the output using the input, process model, and Jacobians as

$$\begin{aligned}\hat{\vec{x}}_{k|k-1} &= f(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, \vec{0}) \\ \hat{\vec{y}}_k &= h(\hat{\vec{x}}_{k|k-1}, \vec{0}) \\ F_{k-1} &= \left[\frac{\partial f}{\partial \vec{x}} \right]_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \\ L_{k-1} &= \left[\frac{\partial f}{\partial \vec{w}} \right]_{\vec{x}_{k-1}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1}=\vec{0}} \\ P_{k|k-1} &= F_{k-1}P_{k-1|k-1}F_{k-1}^T + L_{k-1}Q_{k-1}L_{k-1}^T\end{aligned}\quad (16.75)$$

The discrete-time **IS-EKF correction step** updates the state estimate and the state covariance using the

measurement model and Jacobians as

$$\begin{aligned}
 \tilde{y}_k &= \vec{y}_k - \hat{\vec{y}}_k \\
 H_k &= \left[\frac{\partial h}{\partial \vec{x}} \right]_{\vec{x}_k = \hat{\vec{x}}_{k|k-1}, \vec{v}_k = \vec{0}} \\
 M_k &= \left[\frac{\partial h}{\partial \vec{v}} \right]_{\vec{x}_k = \hat{\vec{x}}_{k|k-1}, \vec{v}_k = \vec{0}} \\
 S_k &= H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \\
 K_k &= P_{k|k-1} H_k^T S_k^{-1} \\
 \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k \text{sat}_{\sigma}(\tilde{y}_k) \\
 P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T \\
 \sigma_{i,k+1} &= \lambda_{1,i} \sigma_{i,k} + \gamma_{1,i} \epsilon_{i,k} \exp(-\epsilon_{i,k}) \\
 \epsilon_{i,k+1} &= \lambda_{2,i} \epsilon_{i,k} + \gamma_{2,i} (\vec{y}_{i,k} - h_{i,k}(\hat{\vec{x}}_{k|k-1}, \vec{0}))^2
 \end{aligned} \tag{16.76}$$

where

$$\text{sat}_{\sigma}(\tilde{y}_k) = \begin{bmatrix} \vdots \\ \text{sat}_{\sqrt{\sigma_i}}(\tilde{y}_i) \\ \vdots \end{bmatrix} \tag{16.77}$$

Second-Order Extended Kalman Filter

Recall that the EKF is based on the first-order Taylor series expansion. An alternative EKF approach is the **second-order extended Kalman filter (SO-EKF)** which truncates the multivariate Taylor series after the quadratic terms, i.e., for some equation $g(\vec{x})$ about the mean of \vec{x} , $\vec{\mu}_x$, as

$$g(\vec{x}) \approx g(\vec{\mu}_x) + G_{\vec{x}}(\vec{x} - \vec{\mu}_x) + \frac{1}{2} \sum_{i=1}^{n_g} \vec{e}_i(\vec{x} - \vec{\mu}_x)^T G_{\vec{x}\vec{x}}^{(i)}(\vec{x} - \vec{\mu}_x) \tag{16.78}$$

where \vec{e}_i is the unit vector for selecting the i^{th} element contribution for $g(\vec{x})$, i.e.,

$$\vec{e}_i = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]^T \tag{16.79}$$

G_x is the Jacobian of $g(\vec{x})$ evaluated at $\vec{x} = \vec{\mu}_x$ as

$$G_x = \left. \frac{\partial g}{\partial \vec{x}} \right|_{\vec{x} = \vec{\mu}_x} \tag{16.80}$$

and $G_{xx}^{(i)}$ is the symmetric Hessian matrix of the $g_i(\vec{x})$ equation for $i = 1, \dots, n_g$ evaluated at $\vec{x} = \vec{\mu}_x$ which has elements at row j and column k as

$$\left[G_{xx}^{(i)} \right]_{j,k} = \left. \frac{\partial^2 g_i}{\partial x_j \partial x_k} \right|_{\vec{x} = \vec{\mu}_x} \tag{16.81}$$

Note that $(\vec{x} - \vec{\mu}_x)^T G_{\vec{x}\vec{x}}^{(i)} (\vec{x} - \vec{\mu}_x)$ is a scalar quantity. Thus, the SO-EKF requires the Hessians must exist for the process and measurement equations and can be more computationally cumbersome to calculate. Notably, higher-order EKFs can also be constructed, but are not commonly used in practice as other nonlinear Kalman filters have been shown to be equivalent with less memory and computational cost.

Computing the expectation with respect to \vec{x} , one can show the mean of $g(\vec{x})$ is approximated as

$$\mathbb{E}[g(\vec{x})] \approx g(\vec{\mu}_x) + \frac{1}{2} \sum_{i=1}^{n_g} \vec{e}_i \text{Tr} \left(G_{xx}^{(i)} P_x \right) \quad (16.82)$$

the cross-covariance of $g(\vec{x})$ and \vec{x} is approximated as

$$\begin{aligned} \mathbb{E}[g(\vec{x})] &= \mathbb{E}[(\vec{x} - \vec{\mu}_x)(g(\vec{x}) - \mathbb{E}[g(\vec{x})])^T] \\ &\approx P_x G_{xx}^{(i)T} \end{aligned} \quad (16.83)$$

and the covariance of $g(\vec{x})$ is approximated as

$$\begin{aligned} \text{Cov}(g(\vec{x})) &= \mathbb{E}[(g(\vec{x}) - \mathbb{E}[g(\vec{x})])(g(\vec{x}) - \mathbb{E}[g(\vec{x})])^T] \\ &\approx G_x P_x G_x^T + \frac{1}{2} \sum_{i=1}^{n_g} \sum_{j=1}^{n_g} \vec{e}_i \vec{e}_j^T \text{Tr} \left(G_{xx}^{(i)} P_x G_{xx}^{(j)} P_x \right) \end{aligned} \quad (16.84)$$

where

$$P_x = \mathbb{E}[(\vec{x} - \vec{\mu}_x)(\vec{x} - \vec{\mu}_x)^T] \quad (16.85)$$

Thus, for the SO-EKF at time step k , one requires the four Jacobians

$$\begin{aligned} F_{k-1} &= \left[\frac{\partial f_{k-1}}{\partial \vec{x}} \right]_{\vec{x}_{k-1} = \hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1} = \vec{0}} \\ L_{k-1} &= \left[\frac{\partial f_{k-1}}{\partial \vec{w}} \right]_{\vec{x}_{k-1} = \hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1} = \vec{0}} \\ H_k &= \left[\frac{\partial h_k}{\partial \vec{x}} \right]_{\vec{x}_k = \hat{\vec{x}}_{k|k-1}, \vec{v}_k = \vec{0}} \\ M_k &= \left[\frac{\partial h_k}{\partial \vec{v}} \right]_{\vec{x}_k = \hat{\vec{x}}_{k|k-1}, \vec{v}_k = \vec{0}} \end{aligned} \quad (16.86)$$

and the $2n_x + 2n_y$ Hessians which have elements at row j and column k as

$$\begin{aligned} \left[F_{xx,k-1}^{(i)} \right]_{j,k} &= \frac{\partial^2 f_{k-1,i}}{\partial x_j \partial x_k} \Bigg|_{\vec{x}_{k-1} = \hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1} = \vec{0}} \quad i = 1, \dots, n_x \\ \left[L_{ww,k-1}^{(i)} \right]_{j,k} &= \frac{\partial^2 f_{k-1,i}}{\partial w_j \partial w_k} \Bigg|_{\vec{x}_{k-1} = \hat{\vec{x}}_{k-1|k-1}, \vec{w}_{k-1} = \vec{0}} \quad i = 1, \dots, n_x \\ \left[H_{xx,k}^{(i)} \right]_{j,k} &= \frac{\partial^2 h_{k,i}}{\partial x_j \partial x_k} \Bigg|_{\vec{x}_k = \hat{\vec{x}}_{k|k-1}, \vec{v}_k = \vec{0}} \quad i = 1, \dots, n_y \\ \left[M_{vv,k}^{(i)} \right]_{j,k} &= \frac{\partial^2 h_{k,i}}{\partial v_j \partial v_k} \Bigg|_{\vec{x}_k = \hat{\vec{x}}_{k|k-1}, \vec{v}_k = \vec{0}} \quad i = 1, \dots, n_y \end{aligned} \quad (16.87)$$

In summary, the discrete-time **SO-EKF initialization step** sets the state estimate and state covariance at $k = 0$ as

$$\begin{aligned} \hat{\vec{x}}_{0|0} &= \hat{\vec{x}}_0 = \mathbb{E} [\vec{x}_0] \\ P_{0|0} &= P_0 = \mathbb{E} [(\vec{x}_0 - \hat{\vec{x}}_{0|0})(\vec{x}_0 - \hat{\vec{x}}_{0|0})^T] \end{aligned} \quad (16.88)$$

The discrete-time **SO-EKF prediction step** predicts the state estimate, the state covariance, and the output using the input, process model, the Jacobians, and the Hessians as

$$\begin{aligned} \hat{\vec{x}}_{k|k-1} &= f_{k-1}(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, \vec{0}) + \frac{1}{2} \sum_{i=1}^{n_x} \vec{e}_i \text{Tr} \left(F_{xx,k-1}^{(i)} P_{k-1|k-1} \right) \\ &\quad + \frac{1}{2} \sum_{i=1}^{n_x} \vec{e}_i \text{Tr} \left(L_{ww,k-1}^{(i)} Q_{k-1} \right) \\ P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \\ &\quad + \frac{1}{2} \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} \vec{e}_i \vec{e}_j^T \text{Tr} \left(F_{xx,k-1}^{(i)} P_{k-1|k-1} F_{xx,k-1}^{(j)} P_{k-1|k-1} \right) \\ &\quad + \frac{1}{2} \sum_{i=1}^{n_x} \sum_{j=1}^{n_x} \vec{e}_i \vec{e}_j^T \text{Tr} \left(L_{xx,k-1}^{(i)} Q_{k-1} L_{xx,k-1}^{(j)} Q_{k-1} \right) \end{aligned} \quad (16.89)$$

The discrete-time **SO-EKF correction step** updates the state estimate and the state covariance using the

measurement model, the Jacobians, and the Hessians as

$$\begin{aligned}
 \tilde{y}_k &= \vec{y}_k - h_k(\hat{\vec{x}}_{k|k-1}, \vec{0}) - \frac{1}{2} \sum_{i=1}^{n_y} \vec{e}_i \text{Tr} \left(H_{ww,k}^{(i)} P_{k|k-1} \right) - \frac{1}{2} \sum_{i=1}^{n_y} \vec{e}_i \text{Tr} \left(M_{ww,k}^{(i)} Q_{k-1} \right) \\
 S_k &= H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \\
 &\quad + \frac{1}{2} \sum_{i=1}^{n_y} \sum_{j=1}^{n_y} \vec{e}_i \vec{e}_j^T \text{Tr} \left(H_{xx,k}^{(i)} P_{k|k-1} H_{xx,k}^{(j)} P_{k|k-1} \right) \\
 &\quad + \frac{1}{2} \sum_{i=1}^{n_y} \sum_{j=1}^{n_y} \vec{e}_i \vec{e}_j^T \text{Tr} \left(M_{xx,k}^{(i)} R_k L_{xx,k}^{(j)} R_k \right) \\
 K_k &= P_{k|k-1} H_k^T S_k^{-1} \\
 \hat{\vec{x}}_{k|k} &= \hat{\vec{x}}_{k|k-1} + K_k \tilde{y}_k \\
 P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T
 \end{aligned} \tag{16.90}$$

Fixed-Interval Extended Kalman Smoother

Analogously to the FI-KF and RTSS, the **fixed-interval extended Kalman smoother (FI-EKS)**, also known as the **extended Rauch-Tung-Striebel smoother (ERTSS)**, assumes one has already approximated the state filtering prior distribution as

$$\vec{X}_{k+1} | \vec{Y}_{1:k} \sim \mathcal{N} \left(\hat{\vec{x}}_{k+1|k}, P_{k+1|k} \right) \tag{16.91}$$

and the state filtering posterior distribution as

$$\vec{X}_k | \vec{Y}_{1:k} \sim \mathcal{N} \left(\hat{\vec{x}}_{k|k}, P_{k|k} \right) \tag{16.92}$$

and will assume the state smoothing posterior distribution to be approximated as

$$\vec{X}_k | \vec{Y}_{1:N} \sim \mathcal{N} \left(\hat{\vec{x}}_{k+1|N}, P_{k+1|N} \right) \tag{16.93}$$

which for $k+1 = N$, one has the initial state smoothing posterior equivalent to the final state filtering posterior, i.e., $\vec{X}_N | \vec{Y}_{1:N} \sim \mathcal{N}(\hat{\vec{x}}_{N|N}, P_{N|N})$.

Then, by similar logic to the previous discussions, the discrete-time **FI-EKS smoothing step** or the **ERTSS smoothing step** updates the state estimate and the state covariance using the posterior state estimate

and covariance and the prior state estimate and covariance or the measurement noise covariance as

$$\begin{aligned}
 \tilde{\vec{x}}_k &= \hat{\vec{x}}_{k+1|N} - \hat{\vec{x}}_{k+1|k} \\
 &= \hat{\vec{x}}_{k+1|N} - f_k(\hat{\vec{x}}_{k|k}, \vec{u}_k, \vec{0}) \\
 F_k &= \left[\frac{\partial f_k}{\partial \vec{x}_k} \right]_{\vec{x}_k=\hat{\vec{x}}_{k|k}, \vec{w}_k=\vec{0}} \\
 L_k &= \left[\frac{\partial f_k}{\partial \vec{w}_k} \right]_{\vec{x}_k=\hat{\vec{x}}_{k|k}, \vec{w}_k=\vec{0}} \\
 K_{S,k} &= P_{k|k} F_k^T P_{k+1|k}^{-1} \\
 &= P_{k|k} F_k^T (F_k P_{k|k} F_k^T + L_k Q_k L_k)^{-1} \\
 \hat{\vec{x}}_{k|N} &= \hat{\vec{x}}_{k|k} + K_{S,k} \tilde{\vec{x}}_k \\
 P_{k|N} &= P_{k|k} + K_{S,k} (P_{k+1|N} - P_{k+1|k}) K_{S,k}^T \\
 &= P_{k|k} + K_{S,k} (P_{k+1|N} - F_k P_{k|k} F_k^T - L_k Q_k L_k) K_{S,k}^T
 \end{aligned} \tag{16.94}$$

References

For more information, please refer to the following

- Fang, H., Haile, M. A., and Wang, Y. “Robust Extended Kalman Filtering for Systems With Measurement Outliers,” in *IEEE Transactions on Control Systems Technology*, 30(2), 2022, pp. 795-802
- Sarkka, S., “5.1 Taylor series expansions,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 64-69
- Sarkka, S., “5.2 Extended Kalman filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 69-75
- Sarkka, S., “8.2 Extended Rauch-Tung-Striebel smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 69-75
- Simon, D., “13.3 Higher-order approaches,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 410-420

16.3 Statistical Linearization and State Estimation

Statistical Linearization

The previous derivations linearized a nonlinear stochastic functions, $g(\vec{X})$, through a Jacobian linearization about a *single* point, e.g., the mean of \vec{X} as

$$\vec{\mu}_x = \mathbb{E} [\vec{X}] \tag{16.95}$$

which provides

$$g(\vec{X}) \approx g(\vec{\mu}_x) + \frac{\partial g(\vec{X})}{\partial \vec{X}} \Big|_{\vec{X}=\vec{\mu}_x} (\vec{X} - \vec{\mu}_x) \quad (16.96)$$

As an alternative, **statistical linearization** optimally fits the linear approximation

$$g(\vec{X}) \approx \vec{b}^{opt} + A^{opt}(\vec{X} - \vec{\mu}_x) \quad (16.97)$$

that minimizes the mean-square error (MSE), i.e.,

$$A^{opt}, \vec{b}^{opt} = \underset{A, \vec{b}}{\operatorname{argmin}} \mathbb{E} \left[\left(g(\vec{X}) - (\vec{b} + A(\vec{X} - \vec{\mu}_x)) \right)^T \left(g(\vec{X}) - (\vec{b} + A(\vec{X} - \vec{\mu}_x)) \right) \right] \quad (16.98)$$

which takes into account the uncertainty or “probabilistic spread” of the random vector, \vec{X} , about its mean through the expectation operator.

To solve this optimization, one can expand the MSE cost function as

$$\text{MSE} = \mathbb{E} \left[g(\vec{X})^T g(\vec{X}) - 2g(\vec{X})^T \vec{b} - 2g(\vec{X})^T A(\vec{X} - \vec{\mu}_x) + \vec{b}^T \vec{b} - 2\vec{b}^T A(\vec{X} - \vec{\mu}_x) + (\vec{X} - \vec{\mu}_x)^T A^T A(\vec{X} - \vec{\mu}_x) \right] \quad (16.99)$$

and performing the expectations on the final two terms, one has

$$\text{MSE} = \mathbb{E} \left[g(\vec{X})^T g(\vec{X}) - 2g(\vec{X})^T \vec{b} - 2g(\vec{X})^T A(\vec{X} - \vec{\mu}_x) + \vec{b}^T \vec{b} \right] + 0 + \text{Tr}(AP_xA^T) \quad (16.100)$$

where P_x is the covariance of \vec{X}

$$P_x = \mathbb{E} [(\vec{X} - \vec{\mu}_x)(\vec{X} - \vec{\mu}_x)^T] \quad (16.101)$$

Then, computing the partial derivatives, one has

$$\frac{\partial \text{MSE}}{\partial \vec{b}} = -2\mathbb{E}[g(\vec{x})] + 2\vec{b} \quad (16.102)$$

and

$$\frac{\partial \text{MSE}}{\partial A} = \mathbb{E}[g(\vec{x})(\vec{X} - \vec{\mu}_x)^T] + 2AP_x \quad (16.103)$$

Finally, solving for the optimal values which provide zero derivatives, one has

$$\vec{b}^{opt} = \mathbb{E}[g(\vec{X})] \quad (16.104)$$

$$A^{opt} = \mathbb{E}[g(\vec{X})(\vec{X} - \vec{\mu}_x)^T] P_x^{-1} \quad (16.105)$$

Thus, the statistical linearization approximation can be written as

$$g(\vec{X}) \approx \mathbb{E}[g(\vec{X})] + \mathbb{E}[g(\vec{X})(\vec{X} - \vec{\mu}_x)^T] P_x^{-1}(\vec{X} - \vec{\mu}_x) \quad (16.106)$$

which has an exact mean, i.e.,

$$g(\vec{X}) \approx \mathbb{E}[g(\vec{X})] \quad (16.107)$$

and an approximate covariance, i.e.,

$$\text{Cov}(g(\vec{X})) = \mathbb{E} [g(\vec{X})(\vec{X} - \vec{\mu}_x)^T] P_x^{-1} \mathbb{E} [g(\vec{X})(\vec{X} - \vec{\mu}_x)^T]^T \quad (16.108)$$

Note that if $g()$ is differentiable, then for multivariate Gaussian random vectors, one has

$$\mathbb{E} [g(\vec{X})(\vec{X} - \vec{\mu}_x)^T] = \mathbb{E} \left[\frac{\partial g(\vec{X})}{\partial \vec{X}} \right] P_x \quad (16.109)$$

which provides the covariance in terms of the expectation of the Jacobian as

$$\text{Cov}(g(\vec{X})) = \mathbb{E} \left[\frac{\partial g(\vec{X})}{\partial \vec{X}} \right] P_x \mathbb{E} \left[\frac{\partial g(\vec{X})}{\partial \vec{X}} \right]^T \quad (16.110)$$

Furthermore, if one can compute the mean expectation, then one can compute the expectation of the Jacobian as the partial derivative, i.e.,

$$\frac{\partial \mathbb{E} [g(\vec{X})]}{\partial \vec{\mu}_x} - \mathbb{E} \left[\frac{\partial g(\vec{X})}{\partial \vec{X}} \right] \quad (16.111)$$

Statistically Linearized (Kalman) Filter

Note that the statistical linearization is useful in Kalman filtering when closed-form solutions exist for these expectations. It is important to note that because statistical linearization takes into account the mean *and* covariance of the random vector and the expectation of the nonlinear functions, the expected linearization error based on statistical linearization tends to be smaller than the Jacobian linearization. However, the statistically linearized filter only applies to multivariate Gaussian noise models. With this previous derivations in mind, the **statistically linearized (Kalman) filter (SLKF)** can be constructed similarly to the EKF. The **SLKF initialization step** sets the state estimate and state covariance at $k = 0$ as

$$\begin{aligned} \hat{x}_{0|0} &= \hat{x}_0 = \mathbb{E} [\vec{X}_0] \\ P_{0|0} &= P_0 = \mathbb{E} [(\vec{X}_0 - \hat{x}_{0|0})(\vec{X}_0 - \hat{x}_{0|0})^T] \end{aligned} \quad (16.112)$$

The discrete-time **SLKF prediction step** predicts the state estimate and the state covariance using the input and process model and explicit expectations as

$$\begin{aligned} \hat{x}_{k|k-1} &= \mathbb{E} [f_{k-1}(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})] \\ P_{k|k-1} &= \mathbb{E} [f_{k-1}(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})(\vec{X}_{k-1} - \hat{x}_{k-1|k-1})^T] P_{k-1|k-1}^{-1} \mathbb{E} [f_{k-1}(\vec{X}, \vec{u}_{k-1}, \vec{W})(\vec{X}_{k-1} - \hat{x}_{k-1|k-1})^T]^T \\ &\quad + \mathbb{E} [f_{k-1}(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1}) \vec{W}_{k-1}^T] Q_{k-1}^{-1} \mathbb{E} [f_{k-1}(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1}) \vec{W}_{k-1}^T]^T \end{aligned} \quad (16.113)$$

where if $f()$ is differentiable, one has

$$\begin{aligned} \hat{x}_{k|k-1} &= \mathbb{E} [f_{k-1}(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})] \\ F_{k-1} &= \mathbb{E} \left[\frac{\partial f_{k-1}(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{X}_{k-1}} \right] \\ L_{k-1} &= \mathbb{E} \left[\frac{\partial f_{k-1}(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{W}_{k-1}} \right] \\ P_{k|k-1} &= F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \end{aligned} \quad (16.114)$$

where the expectations in the prediction step are taken with respect to $\vec{X}_{k-1} \sim \mathcal{N}(\hat{x}_{k-1|k-1}, P_{k-1|k-1})$ and $\vec{W}_{k-1} \sim \mathcal{N}(0, Q_{k-1})$

The discrete-time **SLKF correction step** updates the state estimate and the state covariance using the measurement model and explicit expectations as

$$\begin{aligned}\hat{y}_k &= \mathbb{E}[h(\vec{X}_k, \vec{V}_k)] \\ \tilde{y}_k &= \vec{y}_k - \hat{y}_k \\ S_k &= \mathbb{E}[h_k(\vec{X}_k, \vec{V}_k)(\vec{X}_k - \hat{x}_{k|k-1})^T] P_{k|k-1}^{-1} \mathbb{E}[h_k(\vec{X}_k, \vec{V}_k)(\vec{X}_k - \hat{x}_{k|k-1})^T]^T \\ &\quad + \mathbb{E}[h_k(\vec{X}_k, \vec{V}_k))\vec{V}_k^T] R_k^{-1} \mathbb{E}[h_k(\vec{X}_k, \vec{V}_k))\vec{V}_k^T]^T \\ K_k &= \mathbb{E}[h_k(\vec{X}_k, \vec{V}_k))(\vec{X}_k - \hat{x}_{k|k-1})^T]^T S_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k \\ P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T\end{aligned}\tag{16.115}$$

where if $h_k()$ is differentiable, one has

$$\begin{aligned}\hat{y}_k &= \mathbb{E}[h(\vec{X}_k, \vec{V}_k)] \\ \tilde{y}_k &= \vec{y}_k - \hat{y}_k \\ H_k &= \mathbb{E}\left[\frac{\partial h_k(\vec{X}_k, \vec{V}_k)}{\partial \vec{X}_k}\right] \\ M_k &= \mathbb{E}\left[\frac{\partial h_k(\vec{X}_k, \vec{V}_k)}{\partial \vec{V}_k}\right] \\ S_k &= H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \\ K_k &= P_{k|k-1} H_k^T S_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{y}_k \\ P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T\end{aligned}\tag{16.116}$$

where the expectations in the correction step are taken with respect to $\vec{X}_k \sim \mathcal{N}(\hat{x}_{k|k-1}, P_{k|k-1})$ and $\vec{V}_k \sim \mathcal{N}(0, R_k)$

Statistically Linearized (Kalman) Smoother

Analogously to the FI-EKF/ERTSS, the **fixed-interval statistically linearized (Kalman) smoother (FI-SLKS)**, also known as the **statistically linearized Rauch-Tung-Striebel smoother (SLRTSS)**, assumes one has already approximated the state filtering prior distribution as

$$\vec{X}_{k+1} | \vec{Y}_{1:k} \sim \mathcal{N}\left(\hat{x}_{k+1|k}, P_{k+1|k}\right)\tag{16.117}$$

and the state filtering posterior distribution as

$$\vec{X}_k | \vec{Y}_{1:k} \sim \mathcal{N}\left(\hat{x}_{k|k}, P_{k|k}\right)\tag{16.118}$$

and will assume the state smoothing posterior distribution to be approximated as

$$\vec{X}_k | \vec{Y}_{1:N} \sim \mathcal{N}\left(\hat{\vec{x}}_{k+1|N}, P_{k+1|N}\right) \quad (16.119)$$

which for $k+1 = N$, one has the initial state smoothing posterior equivalent to the final state filtering posterior, i.e., $\vec{X}_N | \vec{Y}_{1:N} \sim \mathcal{N}(\hat{\vec{x}}_{N|N}, P_{N|N})$.

Then, by similar logic to the previous discussions, the discrete-time **FI-SLKS smoothing step** or the **SLRTSS smoothing step** updates the state estimate and the state covariance using the posterior state estimate and covariance and the prior state estimate and covariance or the measurement noise covariance as

$$\begin{aligned} \tilde{\vec{x}}_k &= \hat{\vec{x}}_{k+1|N} - \hat{\vec{x}}_{k+1|k} \\ &= \hat{\vec{x}}_{k+1|N} - \mathbb{E}[f_k(\vec{X}_k, \vec{u}_k, \vec{W}_{k-1})] \\ P_{k+1|k} &= \mathbb{E}\left[\frac{\partial f_k(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{X}_{k-1}}\right] P_{k|k}^{-1} \mathbb{E}\left[\frac{\partial f_k(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{X}_{k-1}}\right]^T \\ &\quad + \mathbb{E}\left[\frac{\partial f_k(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{W}_{k-1}}\right] Q_k^{-1} \mathbb{E}\left[\frac{\partial f_k(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{W}_{k-1}}\right]^T \quad (16.120) \\ K_{S,k} &= \mathbb{E}[f_k(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})(\vec{X}_k - \hat{\vec{x}}_{k|k})^T]^T P_{k+1|k}^{-1} \\ \hat{\vec{x}}_{k|N} &= \hat{\vec{x}}_{k|k} + K_{S,k} \tilde{\vec{x}}_k \\ P_{k|N} &= P_{k|k} + K_{S,k}(P_{k+1|N} - P_{k+1|k})K_{S,k}^T \end{aligned}$$

where if $f_k()$ is differentiable, one has

$$\begin{aligned} \tilde{\vec{x}}_k &= \hat{\vec{x}}_{k+1|N} - \hat{\vec{x}}_{k+1|k} \\ &= \hat{\vec{x}}_{k+1|N} - \mathbb{E}[f_k(\vec{X}_k, \vec{u}_k, \vec{W}_{k-1})] \\ F_k &= \mathbb{E}\left[\frac{\partial f_k(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{X}_{k-1}}\right] \\ L_k &= \mathbb{E}\left[\frac{\partial f_k(\vec{X}_{k-1}, \vec{u}_{k-1}, \vec{W}_{k-1})}{\partial \vec{W}_{k-1}}\right] \quad (16.121) \\ K_{S,k} &= P_{k|k} F_k^T P_{k+1|k}^{-1} \\ &= P_{k|k} F_k^T (F_k P_{k|k} F_k^T + L_k Q_k L_k)^{-1} \\ \hat{\vec{x}}_{k|N} &= \hat{\vec{x}}_{k|k} + K_{S,k} \tilde{\vec{x}}_k \\ P_{k|N} &= P_{k|k} + K_{S,k}(P_{k+1|N} - P_{k+1|k})K_{S,k}^T \\ &= P_{k|k} + K_{S,k}(P_{k+1|N} - F_k P_{k|k} F_k^T - L_k Q_k L_k)K_{S,k}^T \end{aligned}$$

where the expectations in the smoothing step are taken with respect to $\vec{X}_k \sim \mathcal{N}(\hat{\vec{x}}_{k|k}, P_{k|k})$ and $\vec{W}_k \sim \mathcal{N}(0, Q_k)$.

Furthermore, just as a Jacobian linearization is a first-order approximation of the Taylor series expansion, a statistical linearization is a first-order approximation of the Fourier-Hermite series expansion. Thus, a higher-order statistically linearized Kalman filter or smoother is known as a **Fourier-Hermite Kalman**

filter (FHKF) or a **fixed-interval Fourier-Hermite Kalman smoother (FI-FHKS)**, also known as the **Fourier-Hermite Rauch-Tung-Striebel smoother (FHRTSS)**. However, the development of the FHKF and the FI-FHKF/FHRTSS is beyond the scope of this textbook. In summary, they involve expectations of higher-order derivatives with special computational properties due to the Fourier-Hermite series.

Sigma-Point Approximation to Statistical Linearization

The statistical linearization procedure requires explicit computation of the expectation, one can approximate the statistical linearization through a linear regression problem of n_σ points drawn from the prior probability distribution of the random vector, known as the **sigma-points**, and the *true* nonlinear functional evaluation of the sigma-points, i.e., through **statistical linear regression**. However, the exact determination of the *sigma-points* and their corresponding regression weights must be chosen by some criterion of which there are several different methods, e.g., the central divided difference, the unscented transform, the third-order spherical cubature, and Gauss-Hermite quadrature. These are related in the next section on sigma-point Kalman filters and fixed-interval smoothers utilizing these specific methods.

Consider a nonlinear function, $\vec{y} = g(\vec{x})$, which can be evaluated at the $j = 1, \dots, n_\sigma$ sigma-points (\vec{x}_j, \vec{y}_j) where $\vec{y}_j = g(\vec{x}_j)$. Then, one can define

$$\bar{x} = \sum_{j=1}^{n_\sigma} w_j \vec{x}_j \quad (16.122)$$

$$\hat{P}_x = \sum_{j=1}^{n_\sigma} w_j (\vec{x}_j - \bar{x})(\vec{x}_j - \bar{x})^T \quad (16.123)$$

$$\bar{y} = \sum_{j=1}^{n_\sigma} w_j \vec{y}_j \quad (16.124)$$

$$\hat{P}_y = \sum_{j=1}^{n_\sigma} w_j (\vec{y}_j - \bar{y})(\vec{y}_j - \bar{y})^T \quad (16.125)$$

$$\hat{P}_{xy} = \sum_{j=1}^{n_\sigma} w_j (\vec{x}_j - \bar{x})(\vec{y}_j - \bar{y})^T \quad (16.126)$$

where w_j for $j = 1, \dots, n_\sigma$ are a set of scalar **regression weights** such that

$$\sum_{j=1}^{n_\sigma} w_j = 1 \quad (16.127)$$

Thus, for a sigma-point approximation to statistical linearization, i.e.,

$$g(\vec{x}) \approx A\vec{x}_j + \vec{b} \quad (16.128)$$

one minimizes the mean-square errors for a *point-wise* linearization. Here, the point-wise errors are

$$g(\vec{x}_j) - (A\vec{x}_j + \vec{b}) = (\vec{y}_j) - (A\vec{x}_j + \vec{b}) = \quad (16.129)$$

This results in the following optimization problem

$$A^{opt}, \vec{b}^{opt} = \underset{A, \vec{b}}{\operatorname{argmin}} \mathbb{E} \left[\sum_{j=1}^{n_\sigma} w_j \left(g(\vec{X}_j) - (A\vec{X}_j + \vec{b}) \right) \left(g(\vec{X}_j) - (A\vec{X}_j + \vec{b}) \right)^T \right] \quad (16.130)$$

or

$$A^{opt}, \vec{b}^{opt} = \underset{A, \vec{b}}{\operatorname{argmin}} \mathbb{E} \left[\sum_{j=1}^{n_\sigma} w_j \left(\vec{Y}_j - (A\vec{X}_j + \vec{b}) \right) \left(\vec{Y}_j - (A\vec{X}_j + \vec{b}) \right)^T \right] \quad (16.131)$$

which can be solved by considering the partial derivatives of the cost function.

First, rewrite the MSE cost function as

$$\text{MSE} = \mathbb{E} \left[\sum_{j=1}^{n_\sigma} w_j \left((\vec{Y}_j - A\vec{X}_j) - \vec{b} \right) \left((\vec{Y}_j - A\vec{X}_j) - \vec{b} \right)^T \right] \quad (16.132)$$

$$\text{MSE} = \mathbb{E} \left[\sum_{j=1}^{n_\sigma} w_j \left((\vec{Y}_j - A\vec{X}_j)(\vec{Y}_j - A\vec{X}_j)^T - 2(\vec{Y}_j - A\vec{X}_j)\vec{b}^T + \vec{b}\vec{b}^T \right) \right] \quad (16.133)$$

and setting the partial derivative of the cost function with respect to \vec{b} to zero, one has

$$\mathbb{E} \left[\sum_{j=1}^{n_\sigma} 2w_j \left(\vec{Y}_j - A\vec{X}_j - \vec{b} \right) \right] = 0 \quad (16.134)$$

or, by dividing by 2 and the sigma-point statistic definitions, one has

$$\mathbb{E} \left[\bar{Y} - A\bar{X} - \sum_{j=1}^{n_\sigma} w_j \vec{b} \right] = 0 \quad (16.135)$$

which, by $\sum_{j=1}^{n_\sigma} w_j = 1$, provides the optimal solution

$$\vec{b}^{opt} = \bar{Y} - A\bar{X} \quad (16.136)$$

Next, substituting this result for \vec{b} into the cost function, one has

$$A^{opt} = \underset{A}{\operatorname{argmin}} \mathbb{E} \left[\sum_{j=1}^{n_\sigma} w_j \left(\vec{Y}_j - (A\vec{X}_j + \bar{Y} - A\bar{X}) \right) \left(\vec{Y}_j - (A\vec{X}_j + \bar{Y} - A\bar{X}) \right)^T \right] \quad (16.137)$$

$$A^{opt} = \underset{A}{\operatorname{argmin}} \mathbb{E} \left[\sum_{j=1}^{n_\sigma} w_j \left((\vec{Y}_j - \bar{Y}) - A(\vec{X}_j - \bar{X}) \right) \left((\vec{Y}_j - \bar{Y}) - A(\vec{X}_j - \bar{X}) \right)^T \right] \quad (16.138)$$

or

$$A^{opt} = \operatorname{argmin} \mathbb{E} \left[\sum_{j=1}^{n_\sigma} w_j \left((\vec{Y}_j - \bar{Y})(\vec{Y}_j - \bar{Y})^T - 2A(\vec{X}_j - \bar{X})(\vec{Y}_j - \bar{Y}) + A(\vec{X}_j - \bar{X})(\vec{X}_j - \bar{X})^T A^T \right) \right] \quad (16.139)$$

and setting the partial derivative of the cost function with respect to A to zero, one has

$$\mathbb{E} \left[\sum_{j=1}^{n_\sigma} 2w_j \left(A(\vec{X}_j - \bar{X})(\vec{X}_j - \bar{X})^T - (\vec{X}_j - \bar{X})(\vec{Y}_j - \bar{Y}) \right) \right] = 0 \quad (16.140)$$

or, by the sigma-point statistic definitions, one has

$$\mathbb{E} [A \hat{P}_x - \hat{P}_{xy}] = 0 \quad (16.141)$$

which provides the optimal solution

$$A^{opt} = \hat{P}_{xy}^T \hat{P}_x^{-1} \quad (16.142)$$

Substituting for \vec{b} and A , the sigma-point approximation to statistical linearization for a nonlinear function $g()$ is

$$g(\vec{x}) \approx \hat{P}_{xy}^T \hat{P}_x^{-1} (\vec{X}_j) + \left(\bar{Y} - \hat{P}_{xy}^T \hat{P}_x^{-1} \bar{X} \right) \quad (16.143)$$

$$g(\vec{x}) \approx \bar{Y} + \hat{P}_{xy}^T \hat{P}_x^{-1} (\vec{X}_j) - \bar{X} \quad (16.144)$$

or, by substitution, one has

$$g(\vec{x}) \approx \sum_{j=1}^{n_\sigma} w_j g(\vec{X}_j) + \left[\left(\sum_{j=1}^{n_\sigma} w_j (\vec{X}_j - \bar{X}) \left(g(\vec{X})_j - \left(\sum_{j=1}^{n_\sigma} w_j g(\vec{X}_j) \right) \right)^T \right)^T \left(\sum_{j=1}^{n_\sigma} w_j (\vec{X}_j - \bar{X})(\vec{X}_j - \bar{X})^T \right)^{-1} \right] \left(\vec{X}_j - \sum_{j=1}^{n_\sigma} w_j \vec{X}_j \right) \quad (16.145)$$

which demonstrates the numerical integration perspective for the sigma-point method for the expectation integrals of statistical linearization.

References

For more information, please refer to the following

- Sarkka, S., “5.3 Statistical linearization,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 75-77
- Sarkka, S., “5.4 Statistically linearized filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 77-81
- Sarkka, S., “9.2 Statistically linearized Rauch-Tung-Striebel smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 146-148

16.4 Sigma-Point Kalman Filtering and Smoothing

Based on the different types of sigma-point methods, the family of **sigma-point Kalman filters (SPKFs)** includes the Unscented Kalman Filter (UKF), the Central Difference Kalman Filter (CDKF), the Cubature Kalman filter (CKF), and the Gauss-Hermite Kalman filter (GHKF). Similarly, the family of **fixed-interval sigma-point Kalman smoothers (FI-SPKSs)** includes the unscented Kalman smoother (FI-UKS), the central difference Kalman smoother (FI-CDKS), the cubature Kalman smoother (FI-CKS), and the Gauss-Hermite Kalman smoother (FI-GHKS), also known as the family of **sigma-point Rauch-Tung-Striebel smoothers (SPRTSSs)** includes the unscented Rauch-Tung-Striebel smoother (URTSS), the central difference Rauch-Tung-Striebel smoother (CDRTSS), the cubature Rauch-Tung-Striebel smoother (CRTSS), and the Gauss-Hermite Rauch-Tung-Striebel smoother (GHRTSS). This section first presents the general form of an SPKF and an FI-SPKS or SPRTSS, and then defines the different sigma-point methods explicitly for the SPKF and FI-SPKS/SPRTSS. It should be noted that square-root forms of the SPKFs predict and correct the square root of the state covariance directly in Cholesky factored form, using the sigma-point methods and the following three linear algebra techniques: QR decomposition, Cholesky factor updating, and efficient pivot-based least-squares. While the square-root forms of the SPKFs do not display significant accuracy improvement, they have reduced computational complexity for certain nonlinear state-space models as well as increased numerical stability. Lastly, it should be noted that for *very* large state dimensions, the SPKFs can be computationally faster than the EKF since no Jacobian matrices must be calculated and no matrix inversion of the state dimension is necessary.

Sigma-Point Kalman Filters

The discrete-time **SPKF initialization step** sets the initial state estimate as

$$\hat{\vec{x}}_{0|0} = \hat{\vec{x}}_0 = \mathbb{E} [\vec{x}_0] \quad (16.146)$$

the initial covariance as

$$P_{0|0} = P_0 = \mathbb{E} [(\vec{x}_0 - \hat{\vec{x}}_0)(\vec{x}_0 - \hat{\vec{x}}_0)^T] \quad (16.147)$$

The discrete-time **SPKF prediction step** first computes $j = 1, \dots, n_{a,p}$ new **prediction augmented sigma-points** as

$$\vec{\chi}_{j,k-1|k-1}^a = \hat{\vec{x}}_{k-1|k-1}^a + \sqrt{P_{k-1|k-1}^a} \vec{\xi}_j \quad (16.148)$$

where $\vec{\xi}_j$ depends on the sigma-point method chosen, the **SPKF process augmented state estimate** of dimension $n_\xi = n_x + n_w$ is defined as

$$\hat{\vec{x}}_{k-1|k-1}^a = \begin{bmatrix} \hat{\vec{x}}_{k-1|k-1} \\ \vec{0} \end{bmatrix} \quad (16.149)$$

the **SPKF augmented state covariance** of dimension $n_\xi \times n_\xi$ is defined as

$$P_{k-1|k-1}^a = \begin{bmatrix} P_{k-1|k-1} & 0 \\ 0 & Q_{k-1} \end{bmatrix} \quad (16.150)$$

Then, one uses these augmented state sigma-points to compute the prior state sigma-points, the prior state estimate, and the prior state covariance as

$$\begin{aligned}\hat{\vec{X}}_{j,k|k-1} &= f_{k-1}(\left[\vec{X}_{j,k-1}\right]_{1:n_x}, \vec{u}_{k-1}, \left[\vec{X}_{j,k-1}\right]_{n_x+1:n_\xi}) \\ \hat{\vec{x}}_{k|k-1} &= \sum_{j=1}^{n_{a,p}} w_j^{(m)} \hat{\vec{X}}_{j,k|k-1} \\ P_{k|k-1} &= \sum_{j=1}^{n_{a,p}} w_j^{(c)} (\hat{\vec{X}}_{j,k|k-1} - \hat{\vec{x}}_{k|k-1})(\hat{\vec{X}}_{j,k|k-1} - \hat{\vec{x}}_{k|k-1})^T\end{aligned}\quad (16.151)$$

where $[\vec{\bullet}]_{1:n}$ selects the sub-vector of elements 1 through n of $\vec{\bullet}$ and $w_j^{(m)}$ is the j^{th} mean weight and $w_j^{(c)}$ is the j^{th} covariance weight corresponding to the j^{th} sigma-point.

The discrete-time **SPKF correction step** recomputes $n_{a,c}$ new **correction augmented sigma-points** as

$$\vec{X}_{j,k|k-1} = \hat{\vec{x}}_{k|k-1}^a + \sqrt{P_{k|k-1}^a} \vec{\xi}_j \quad (16.152)$$

where $\vec{\xi}_j$ depends on the sigma-point method chosen, the **SPKF process augmented state estimate** with dimension $n_\xi = n_x + n_v$ is defined as

$$\hat{\vec{x}}_{k|k-1}^a = \begin{bmatrix} \hat{\vec{x}}_{k|k-1} \\ \vec{0} \end{bmatrix} \quad (16.153)$$

and the **SPKF augmented state covariance** with dimension $n_\xi \times n_\xi$ is defined as

$$P_{k|k-1}^a = \begin{bmatrix} P_{k|k-1} & 0 \\ 0 & R_k \end{bmatrix} \quad (16.154)$$

Then, one uses these augmented state sigma-points to compute the measurement sigma-points, the predicted measurement, the innovation, the innovation covariance, the state-measurement cross-covariance, the Kalman gain, the posterior state estimate, and the posterior state covariance as

$$\begin{aligned}\hat{\vec{y}}_{j,k} &= h_k \left(\left[\vec{X}_{j,k|k-1}^a \right]_{1:n_x}, \left[\vec{X}_{j,k|k-1}^a \right]_{n_x+1:n_\xi} \right) \\ \hat{\vec{y}}_k &= \sum_{j=1}^{n_{a,c}} w_j^{(m)} \hat{\vec{y}}_{j,k} \\ \tilde{\vec{y}}_k &= \vec{y}_k - \hat{\vec{y}}_k \\ S_k &= \sum_{j=0}^{n_{a,c}} w_j^{(c)} \left(\hat{\vec{y}}_{j,k} - \hat{\vec{y}}_k \right) \left(\hat{\vec{y}}_{j,k} - \hat{\vec{y}}_k \right)^T \\ P_{xy} &= \sum_{j=0}^{n_{a,c}} w_j^{(c)} \left(\left[X_{j,k|k-1}^a \right]_{1:n_x} - \hat{x}_{k|k-1} \right) \left(\hat{\vec{y}}_{j,k} - \hat{\vec{y}}_k \right)^T \\ K_k &= P_{xy} S_k^{-1} \\ \hat{x}_{k|k} &= \hat{x}_{k|k-1} + K_k \tilde{\vec{y}}_k \\ P_{k|k} &= P_{k|k-1} - K_k S_k K_k^T\end{aligned}\quad (16.155)$$

Sigma-Point Kalman Smoother

The discrete-time **FI-SPKS smoothing step**, also known as the **SPRTSS smoothing step**, first recomputes $n_{a,s}$ new **correction augmented sigma-points** as

$$\vec{\chi}_{j,k|k} = \hat{\vec{x}}_{k|k} + \sqrt{P_{k|k}} \vec{\xi}_j \quad (16.156)$$

where $\vec{\xi}_j$ depends on the sigma-point method chosen, the **FI-SPKS/SPRTSS process augmented state estimate** with dimension $n_\xi = n_x + n_w$ is defined as

$$\hat{\vec{x}}_{k|k}^a = \begin{bmatrix} \hat{\vec{x}}_{k|k} \\ \vec{0} \end{bmatrix} \quad (16.157)$$

and the **FI-SPKS/SPRTSS augmented state covariance** with dimension $n_\xi \times n_\xi$ is defined as

$$P_{k|k}^a = \begin{bmatrix} P_{k|k} & 0 \\ 0 & Q_k \end{bmatrix} \quad (16.158)$$

Then, one uses the augmented state sigma-points through the exact nonlinear measurement equation to obtain the measurement sigma-points as

$$\begin{aligned} \hat{\vec{x}}_{j,k+1|k} &= f_k(\left[\vec{\chi}_{j,k}\right]_{1:n_x}, \vec{u}_k, \left[\vec{\chi}_{j,k}\right]_{n_x+1:n_\xi}) \\ \hat{\vec{x}}_{k+1|k} &= \sum_{j=1}^{n_{a,s}} w_j^{(m)} \hat{\vec{x}}_{j,k+1|k} \\ \tilde{\vec{x}}_{k+1} &= \hat{\vec{x}}_{k+1|N} - \hat{\vec{x}}_{k+1|k} \\ P_{k+1|k} &= \sum_{j=0}^{n_{a,s}} w_j^{(c)} \left(\hat{\vec{x}}_{j,k+1|k} - \hat{\vec{x}}_{k+1|k} \right) \left(\hat{\vec{x}}_{j,k+1|k} - \hat{\vec{x}}_{k+1|k} \right)^T \\ P_{k,k+1} &= \sum_{j=0}^{n_{a,s}} w_j^{(c)} \left(\left[\vec{\chi}_{j,k} \right]_{1:n_x} - \hat{\vec{x}}_{k|k} \right) \left(\hat{\vec{x}}_{j,k+1|k} - \hat{\vec{x}}_{k+1|k} \right)^T \\ K_{S,k} &= P_{k,k+1} P_{k+1|k}^{-1} \\ \hat{\vec{x}}_{k|N} &= \hat{\vec{x}}_{k|k} + K_{S,k} \tilde{\vec{x}}_{k+1} \\ P_{k|N} &= P_{k|k} + K_{S,k} (P_{k+1|N} - P_{k+1|k}) K_{S,k}^T \end{aligned} \quad (16.159)$$

Unscented Transform

One method for determining the sigma-points and corresponding weights is such that they match the moments of the prior random state vector, \vec{X} . This is achieved by choosing the sigma-points according to some constraint equation of the form $\eta(\{\mathcal{X}, w\}, n_\sigma, f_X(\vec{x})) = 0$ where $\{\mathcal{X}, w\}$ is the set of all sigma-points X_j and corresponding weights w_j for $j = 1, \dots, n_\sigma$. It is possible to satisfy such a constraint and still have some degree of freedom in the choice of the sigma-point locations, thus, one typically also uses a cost function, e.g. some $c(\{\mathcal{X}, w\}, n_\sigma, f_X(\vec{x}))$, to incorporate statistical features of \vec{X} which are desirable, but

do not necessarily have to be met. Taken together, one can construct this selection method as the following optimization problem.

$$\begin{aligned} \{\vec{X}, w\} &= \operatorname{argmin} c(\{\vec{X}, w\}, n_\sigma, f_{\vec{X}}(\vec{x})) \\ \text{Subject to } &\eta(\{\vec{X}, w\}, n_\sigma, f_{\vec{X}}(\vec{x})) = 0 \end{aligned} \quad (16.160)$$

Using this general optimization, the **unscented transform (UT)** specifies that the sigma-points should be constrained to at least capture the first and second-order moments of \vec{X} . It can be shown that the *minimum* number of sigma-points required to do so is $2n_x + 1$ and matches the first and second moments accurately for \vec{Y} and \hat{P}_{yy} . It should be noted that additional sigma-points in **higher-order unscented transforms** can be used to capture higher-order moments of \vec{X} . Consideration of these higher-order UTs is beyond the scope of this textbook.

The UKF and FI-UKS/URTSS use the UT to obtain the $n_\sigma = 2n_\xi + 1$ sigma-points in the SPKF framework. For the first sigma-point of the UT, one has

$$\vec{\xi}_1 = \vec{0} \quad (16.161)$$

with mean weight

$$w_1^{(m)} = \frac{\lambda}{\lambda + n_\xi} \quad (16.162)$$

and covariance weight

$$w_1^{(c)} = \frac{\lambda}{\lambda + n_\xi} + (1 - \alpha^2 + \beta) \quad (16.163)$$

where $\lambda = \alpha^2(n_\xi + \kappa) - n_\xi$ is a scaling parameter, α determines the spread of the sigma-points about \hat{x} and usually set to $1e^{-2} \leq \alpha \leq 1$, κ is a secondary scaling parameter usually set to either 0 or $3 - n_\xi$, and β is an extra degree of freedom parameter used to incorporate prior information.

For $j = 1, \dots, n_\xi$ in the UT, one has

$$\begin{aligned} \vec{\xi}_{1+j, k-1} &= \sqrt{\lambda + n_\xi} \vec{e}_j \\ \vec{\xi}_{1+j+n_\xi, k-1} &= -\sqrt{\lambda + n_\xi} \vec{e}_j \end{aligned} \quad (16.164)$$

where \vec{e}_j is the unit vector for selecting the j^{th} element contribution for $\vec{\xi}$, i.e.,

$$\vec{e}_j = [0 \quad \cdots \quad 0 \quad 1 \quad 0 \quad \cdots \quad 0]^T \quad (16.165)$$

with mean and covariance weights for $j = 1, \dots, 2n_\xi$ as

$$w_{1+j}^{(m)} = w_{1+j}^{(c)} = \frac{1}{2(\lambda + n_\xi)} \quad (16.166)$$

Central Divided Difference

Another method to determine the sigma-points was derived through the **second-order Stirling's formula** which is computed by replacing the analytically derived first and second order derivatives in the Taylor

series expansion by numerically evaluated **central divided differences (CDD)** which are defined for some nonlinear function $g(\vec{x})$ as

$$\nabla g \approx \frac{g(\vec{x} + h\vec{\delta}_x) - g(\vec{x} - h\vec{\delta}_x)}{2h} \quad (16.167)$$

$$\nabla^2 g \approx \frac{g(\vec{x} + h\vec{\delta}_x) + g(\vec{x} - h\vec{\delta}_x) - 2g(\vec{x})}{h^2} \quad (16.168)$$

where h is the central difference half-step size. For functions of multivariate Gaussian random vectors, it should be noted that $h = \sqrt{3}$ is optimal.

The CDKF and FI-CDKS/CDRTSS use the CDD to obtain the $n_\sigma = 2n_\xi + 1$ sigma-points in the SPKF framework. For the first sigma-point in the CDD, one has

$$\vec{\xi}_1 = \vec{0} \quad (16.169)$$

with mean and covariance weight

$$w_1^{(m)} = w_1^{(c)} = \frac{h^2 - n_\xi}{h^2} \quad (16.170)$$

and for $j = 1, \dots, n_\xi$ in the CDD, one has

$$\begin{aligned} \vec{\xi}_{1+j,k-1} &= h\vec{e}_j \\ \vec{\xi}_{1+j+n_\xi,k-1} &= -h\vec{e}_j \end{aligned} \quad (16.171)$$

with mean and covariance weights for $j = 1, \dots, 2n_\xi$ as

$$w_{1+j}^{(m)} = w_{1+j}^{(c)} = \frac{1}{2h^2} \quad (16.172)$$

By inspection, one can see that the UT is equivalent to the CDD if one chooses $h = \sqrt{3}$, $\alpha = 1$, $\beta = 0$, and $\kappa = 3 - n_\xi$. This choice of κ allows for exact matching of fourth-order monomials for $g()$, but for $n_\xi > 3$, one has negative weights for the first sigma-point which can make the UKF and the CDKF with $h = \sqrt{3}$ unstable.

Third-Order Spherical Cubature

A third method to determine the sigma-points is through **third-order spherical cubature integration rule** which forms a sigma-point approximation of the Gaussian expectation integral for a nonlinear function $g(\vec{x})$ as

$$\int_{-\infty}^{\infty} \cdots g(\vec{\xi}) \mathcal{N}(\vec{\xi}; \vec{0}, I) d\vec{\xi} \approx w \sum_{i=1}^{2n_\xi} g(c\vec{\xi}_i) \quad (16.173)$$

where $\vec{\xi}_i$ belong to the symmetric set **[1]** with generator $[1, 0, \dots, 0]^T$, i.e.,

$$[1] = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots, \begin{bmatrix} -1 \\ 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}, \dots \right\} \quad (16.174)$$

and w and c are chosen to make the integration exact up to third-order monomials for $g()$, i.e.,

$$w = \frac{1}{2n_\xi} \quad (16.175)$$

and

$$c = \sqrt{n_\xi} \quad (16.176)$$

The CKF and FI-CKS/CRTSS use the TOSCR to obtain the $n_\sigma = 2n_\xi$ sigma-points in the SPKF framework. For $j = 1, \dots, n_\xi$ in the TOSCR, one has

$$\begin{aligned} \vec{\xi}_{1+j, k-1} &= \sqrt{n_\xi} \vec{e}_j \\ \vec{\xi}_{1+j+n_\xi, k-1} &= -\sqrt{n_\xi} \vec{e}_j \end{aligned} \quad (16.177)$$

with mean and covariance weights for $j = 1, \dots, 2n_\xi$ as

$$w_{1+j}^{(m)} = w_{1+j}^{(c)} = \frac{1}{2n_\xi} \quad (16.178)$$

By inspection, one can see that the TOSCR is equivalent to the UT if one chooses $\alpha = 1$ and $\beta = \kappa = 0$ which notably discards the first sigma-point in the UT, i.e., sets the weights to zero which was the original version of the UT.

Gauss-Hermite Quadrature

Related to the spherical cubature, a fourth method to determine the sigma-points is through the **Gauss-Hermite quadrature rule (GHQR)** which forms a p^{th} -order sigma-point approximation of the univariate Gaussian expectation integral for a function $g(x)$ as

$$\int_{-\infty}^{\infty} g(\xi) \mathcal{N}(\xi; 0, 1) d\xi \approx \sum_{i=1}^p w_i g(\xi_i) \quad (16.179)$$

where the sigma-points ξ_i are chosen to enforce this quadrature rule exactly for polynomials up to order $2p - 1$. It can be shown that this corresponds to choosing ξ_i as the roots of the p^{th} -order Hermite polynomial, $H_p(\xi)$, given as

$$H_p(\xi) = (-1)^p \exp\left(\frac{\xi^2}{2}\right) \frac{d^p}{d\xi^p} \exp\left(-\frac{\xi^2}{2}\right) \quad (16.180)$$

which can also be computed through the recursion

$$\begin{aligned} H_0(\xi) &= 1 \\ H_1(\xi) &= \xi \\ H_2(\xi) &= \xi^2 - 1 \\ &\vdots = \vdots \\ H_{p+1}(\xi) &= \xi H_p(\xi) - p H_{p-1}(\xi) \end{aligned} \quad (16.181)$$

In practice, these are typically calculated as the eigenvalues of a suitable tri-diagonal matrix.

Then, noting through the transformation of the Gaussian expectation for $X \sim \mathcal{N}(\mu, P)$, one has for the GHQR

$$\int_{-\infty}^{\infty} g(x) \mathcal{N}(x; \mu, P) dx = \int_{-\infty}^{\infty} g(\mu + \sqrt{P}\xi) \mathcal{N}(\xi; 0, 1) d\xi \approx \sum_{i=1}^p w_i g(\mu + \sqrt{P}\xi_i) \quad (16.182)$$

where the unit sigma-points, ξ_i , $i = 1, \dots, p$, are the roots of the p^{th} -order Hermite polynomial and corresponding weights as

$$w_i = \frac{p!}{p^2 [H_{p-1}(\xi_i)]^2} \quad (16.183)$$

Then, for the multivariate random vectors, one can generalize to the p^{th} -order **Gauss-Hermite cubature rule (GHCR)** for a n_x -dimensional Gaussian random vector $\vec{X} \sim \mathcal{N}(\vec{\mu}, P)$ as

$$\int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} g(\vec{x}) \mathcal{N}(\vec{x}; \vec{\mu}, P) d\vec{x} \approx \sum_{k=1}^{n_x} w_{\mathcal{I}_k} g(\vec{\mu} + \sqrt{P} \vec{\xi}_{\mathcal{I}_k}) \quad (16.184)$$

where

$$\mathcal{I}_k = \{i_1, \dots, i_{n_x}\} \quad (16.185)$$

$$\vec{\xi}_{\mathcal{I}_k} = \begin{bmatrix} \xi_{i_1} \\ \vdots \\ \xi_{i_n} \end{bmatrix} \quad (16.186)$$

and the weights are

$$w_{\mathcal{I}_k} = w_{i_1} \cdots w_{i_{n_x}} = \frac{p!}{p^2 [H_{p-1}(\xi_{i_1})]^2} \cdots \frac{p!}{p^2 [H_{p-1}(\xi_{i_n})]^2} \quad (16.187)$$

and ξ_{i_k} , $i_k = 1, \dots, p$ for all possible sets of index combinations, $k = 1, \dots, n_x$, which are the i_k^{th} root of the p^{th} -order Hermite polynomial and w_{i_k} are the corresponding weights.

Thus, the GHKF and FI-GHKS/GHRTSS, also known as the **quadrature Kalman filter (QKF)**, the **fixed-interval quadrature Kalman smoother (FI-QKS)** or the **quadrature Rauch-Tung-Striebel smoother (QRTSS)**, use the p^{th} -order GHCR to obtain the p^{n_x} sigma-points in the SPKF framework. The GHCR uses p^{n_x} sigma-points in order to approximate higher-order polynomials while the UT/CDD/TOSCR only approximate monomials up to the third-order, although by choosing $\kappa = 3 - n_\xi$ and $\beta = 2$, the UT can match fourth-order monomials. This makes the p^{th} -order GHCR covariance approximation exact up to p^{th} -order monomials for $g()$ while for the UT/CDD/TOSCR, the covariance approximations are exact for first-order monomials for $g()$ and maybe the second-order monomials for $g()$ for UT if $\kappa = 3 - n_\xi$ and $\beta = 2$.

References

For more information, please refer to the following

- Sarkka, S., “5.6 Unscented Kalman filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 86-92

- Sarkka, S., “6.4 Gauss-Hermite Kalman filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 103-106
- Sarkka, S., “6.6 Cubature Kalman filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 110-114
- Sarkka, S., “9.3 Unscented Rauch-Tung-Striebel smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 144-146
- Sarkka, S., “10.1 General Gaussian Rauch-Tung-Striebel smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 154-155
- Sarkka, S., “10.2 Gauss-Hermite Rauch-Tung-Striebel smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 155-156
- Sarkka, S., “10.3 Cubature Rauch-Tung-Striebel smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 156-159
- Simon, D., “14.3 Unscented Kalman filtering,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 447-457
- Van Der Merwe, R., “Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models,” in *Oregon Health & Science University*, 2004
- Wu, Y., Wu, M., and Hu, D., “A Numerical-Integration Perspective on Gaussian Filters,” in *IEEE Transactions on Signal Processing*, 2006, pp. 2910-2921

16.5 Particle Filtering

Nonlinear Kalman filters assume a multivariate Gaussian posterior conditional PDF for $\vec{X}_k | \vec{y}_{1:k}$. Thus, this can be a poor approximation of the optimal Bayes filter when $\vec{X}_k | \vec{y}_{1:k}$ is **multi-modal**, i.e. there are multiple values for the maximum of $f_{\vec{X}_k | \vec{Y}_{1:k}}(\vec{x} | \vec{y}_{1:k})$, and/or $\vec{X} | \vec{Y}_{1:k}$ is **heavy-tailed**, i.e. one cannot bound $\vec{X} | \vec{Y}_{1:k}$ by an exponential function as $\vec{x} \rightarrow \infty$.

While advanced filters exist for modeling different approximations for the posterior conditional PDFs in the Bayes filter context, the most general approach is to recursively update the posterior distribution using sequential Monte Carlo (SMC) methods by approximating the posterior conditional PDF by a set of weighted samples, i.e. “particles,” without making any analytical assumptions about its form. Thus, one can consider these particles as a PMF approximation to the PDF. Any filter which uses a particle approximation is known as a **particle filter (PF)** and can be used for state estimation for nonlinear, non-Gaussian stochastic state-space systems. This section will introduce the basic concepts used in particle filters.

Sequential Importance Sampling

The primary method in particle filtering is the use of **sequential importance sampling (SIS)** as an extension of Monte Carlo methods of estimation. Recall that importance sampling represents a PDF $f_{\vec{X}}(\vec{x})$ by an

empirical approximation

$$f_{\vec{X}}(\vec{x}) \approx \hat{f}_{\vec{X}}(\vec{x}) = \sum_{j=1}^{n_p} w^{(i)} \delta(\vec{x} - \vec{x}^{(i)}) \quad (16.188)$$

where $\delta(\bullet)$ is the Dirac delta function and $\vec{x}^{(i)}$ for $i = 1, \dots, n_p$ are the particles with corresponding **importance weights**, $w^{(i)}$, which are drawn from some related, easy-to-sample **importance function**, $\pi(\vec{x})$, also known as the **proposal PDF**. With this definition, the importance weights are given by

$$w^{(i)} = \frac{\frac{f_{\vec{X}}(\vec{x}^{(i)})}{\pi(\vec{x}^{(i)})}}{\sum_{i=1}^{n_p} \frac{f_{\vec{X}}(\vec{x}^{(i)})}{\pi(\vec{x}^{(i)})}} \quad (16.189)$$

Furthermore, the expectation of the function, $g(\vec{x})$, can be approximated using this PMF approximation, e.g.

$$\mathbb{E}[g(\vec{x})] = \int g(\vec{x}) f_{\vec{X}}(\vec{x}) d\vec{x} \approx \sum_{i=1}^{n_p} w^{(i)} g(\vec{x}^{(i)}) \quad (16.190)$$

In the Bayes filter context with states $\vec{x}_{1:k}$ and measurements, $\vec{y}_{1:k}$, if one assumes HMM and conditional independence between the state and measurements, then one can write the importance function as a recursive relationship

$$\pi(\vec{x}_{0:k} | \vec{y}_{1:k}) = \pi(\vec{x}_{1:k-1} | \vec{y}_{1:k-1}) \pi(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k}) \quad (16.191)$$

which allows the recursive computation of the importance weights as

$$w_k^{(i)} \propto \frac{f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k^{(i)}) f_{\vec{X}}(\vec{x}_k^{(i)} | \vec{x}_{k-1}^{(i)})}{\pi(\vec{x}_k^{(i)} | \vec{x}_{0:k-1}, \vec{y}_{1:k})} w_{k-1}^{(i)} \quad (16.192)$$

which allows one to sequentially update the importance weights given an appropriate choice of importance function, $\pi(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k})$.

For the Bayes filter, recall that one can sample from this importance function, evaluate the likelihood function, $f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k)$, and evaluate the state transition PDF, $f_{\vec{X}}(\vec{x}_k | \vec{x}_{k-1})$, one can iteratively compute the importance weights if one generates an prior set of particles and importance weights. This procedure allows one to compute

$$\mathbb{E}[g(\vec{x})] \approx \frac{\frac{1}{n_p} \sum_{i=1}^{n_p} w_k^{(i)} g(\vec{x}_k^{(i)})}{\frac{1}{n_p} \sum_{i=1}^{n_p} w_k^{(i)}} = \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} g(\vec{x}_k^{(i)}) \quad (16.193)$$

with **normalized importance weights**

$$\tilde{w}_k^{(i)} \propto \frac{w_k^{(i)}}{\sum_{i=1}^{n_p} w_k^{(i)}} \quad (16.194)$$

where it should be noted that this estimate asymptotically converges if the mean and variance of $g(\vec{x}_k)$ and w_k exist and are bounded and if the support of the importance function includes the support of the posterior PDF. Thus, as $n_p \rightarrow \infty$, the posterior conditional PDF of the Bayes filter can be approximately by the PMF estimate

$$\hat{f}_{\vec{X}_k | \vec{Y}_{1:k}}(\vec{x}_k | \vec{y}_{1:k}) \approx \sum_{i=1}^{n_p} \tilde{w}^{(i)} \delta(\vec{x}_k - \vec{x}_k^{(i)}) \quad (16.195)$$

where the approximation improves as the number of particles is increased.

Then, note that one can then form a state estimate based on a Bayes estimator, e.g., the MAP state estimate

$$\hat{\vec{x}}_{PF,MAP} = \vec{x}^{(i)} \text{ with } i \text{ chosen as } \operatorname{argmax}_i \tilde{w}^{(i)} \quad (16.196)$$

or the MMSE estimate

$$\hat{\vec{x}}_{PF,MMSE} = \sum_{i=1}^{n_p} \tilde{w}^{(i)} \vec{x}_k^{(i)} \quad (16.197)$$

and one can also estimate other statistics of the posterior conditional PDF, e.g. the covariance as

$$\hat{P}_{PF} = \sum_{i=1}^{n_p} \tilde{w}^{(i)} \left(\vec{x}_k^{(i)} - \hat{\vec{x}}_{PF,MMSE} \right) \left(\vec{x}_k^{(i)} - \hat{\vec{x}}_{PF,MMSE} \right)^T \quad (16.198)$$

Particle Filter Implementation

The most important design criteria in particle filter implementation is the choice of the importance PDF. The optimal importance function which minimizes the variance on the importance weights is given by the true conditional PDF given the *entire* previous state history and all observations, i.e.,

$$\pi^{opt}(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k}) = f_{\vec{X}_k | \vec{X}_{k-1}, \vec{Y}_k}(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k}) \quad (16.199)$$

However, sampling from this conditional PDF may be impractical for arbitrary PDFs for dynamical systems. Thus, other practical choices are usually made for the importance PDF in particle filter implementation.

The simplest choice is the state transition conditional PDF, i.e.

$$\pi(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k}) = f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k | \vec{x}_{k-1}) \quad (16.200)$$

which allows one to form the importance weight recursion as simply

$$w_k^{(i)} \propto f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k^{(i)}) w_{k-1}^{(i)} \quad (16.201)$$

where $f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k^{(i)})$ is the likelihood function.

For example, if the state equation has zero-mean AWGN, \vec{w}_k , with covariance, Q_{k-1} , i.e.

$$\vec{x}_k = f(\vec{x}_{k-1}, \vec{u}_{k-1}) + \vec{w}_k \quad (16.202)$$

then one can choose

$$\pi(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k}) = f_N(\vec{x}_k; f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}), Q_{k-1}) \quad (16.203)$$

and if the measurement equation has zero-mean AWGN, \vec{v}_k , with covariance, R_k , i.e.

$$\vec{y}_k = h(\vec{x}_k) + \vec{v}_k \quad (16.204)$$

then one can choose

$$f_{\vec{Y}_k | \vec{X}_k^{(i)}}(\vec{y}_k | \vec{x}_k^{(i)}) = f_N(\vec{y}_k; h(\vec{x}_k), R_k) \quad (16.205)$$

In addition, one of the primary disadvantages of implementing the SIS algorithm discussed previously is that the variance of the importance weights increases stochastically over time. Typically, after a few iterations, most of the weights tend towards zero, effectively being removed from the particle set due to their numerically insignificance weight. To avoid this **degeneracy problem** of the particles, a resampling stage is used to eliminate particles with low importance weights and multiply samples with high importance weights. These methods are collectively known as **sequential importance sampling with resampling (SIS/R)** methods, also known as the shortened **sequential importance resampling (SIR)** or **sampling importance resampling (SIR)**. Practical methods for this resampling step are another implementation choice in particle filters. Regardless of the chosen resampling method, one should obtain n_p particles, $\tilde{w}^{(m)+}$, distributed approximately according to the posterior PDF.

The simplest method for resampling is the **multinomial resampling** which can be summarized as

$$\vec{x}^{(i)} = \vec{x}^{(j)} \quad \text{with probability } \tilde{w}^{(i)} \quad (16.206)$$

where it can be shown that the ensemble PDF of the resampled particles in this method tends to the posterior conditional PDF as $n_p \rightarrow \infty$. Thus, one has the approximation

$$f_{\vec{X}_k | \vec{Y}_k}(\vec{x}_k | \vec{y}_{1:k}) \approx \hat{f}_{X|Y}(\vec{x}_k | \vec{y}_{1:k}) = \frac{1}{n_p} \sum_{i=1}^{n_p} \delta(\vec{x}_k - \vec{x}_k^{(i)}) \quad (16.207)$$

where the PDF has been approximated by number of copies instead of the importance weights. Thus, one no longer recursively updates the weights, but has the model

$$w_k^{(i)} \propto \frac{f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k^{(i)}) f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k^{(i)} | \vec{x}_{k-1}^{(i)})}{\pi(\vec{x}_k^{(i)} | \vec{x}_{0:k-1}^{(i)}, \vec{y}_{1:k})} \quad (16.208)$$

This can be written out as the following algorithm.

For $i = 1, \dots, n_p$

Sample $r \sim \mathcal{U}[0, 1]$

If $\sum_{m=1}^{j-1} \tilde{w}_k^{(m)} < r$ but $\sum_{m=1}^j \tilde{w}_k^{(m)} \geq r$, set $\vec{x}^{(i)} = \vec{x}^{(j)}$

Residual resampling, also known as **remainder resampling**, attempts to decrease the variance of the SIR by keeping copies of particles based on rounding down the expected number of particles based on the importance weights, then only resampling based on the remainders from all particles. This can be written out as the following algorithm.

Set $n_c = 1$

Set $n_r = n_p$

For $i = 1, \dots, n_p$

For $j = 1, \dots, \text{floor}(n_p \tilde{w}_k^{(i)})$

Set $\vec{x}^{(n_c)} = \vec{x}^{(i)}$
 $n_c = n_c + 1$
 $n_r = n_r - \text{floor}(n_p \tilde{w}_k^{(i)})$

For $i = 1, \dots, n_r$

Sample $r \sim \mathcal{U}[0, 1]$
If $\sum_{m=1}^{j-1} n_p \tilde{w}_k^{(m)} - \text{floor}(n_p \tilde{w}_k^{(i)}) < r$ but $\sum_{m=1}^j n_p \tilde{w}_k^{(m)} - \text{floor}(n_p \tilde{w}_k^{(i)}) \geq r$, set $\vec{x}^{(n_c)} = \vec{x}^{(j)}$
 $n_c = n_c + 1$

Stratified resampling pre-partitions the $[0, 1]$ interval into n_p disjoint sets and samples from these. Then, one performs the resampling, i.e.

For $i = 1, \dots, n_p$

Sample $r(i) \sim \mathcal{U}[(i-1)/n_p, i/n_p]$
If $\sum_{m=1}^{j-1} \tilde{w}_k^{(m)} < r(i)$ but $\sum_{m=1}^j \tilde{w}_k^{(m)} \geq r(i)$, set $\vec{x}^{(i)} = \vec{x}^{(j)}$

This approach can be even further simplified by **systematic resampling**, a.k.a. **low-variance resampling (LVR)**, also known as the **stochastic universal sampler**, which samples a *single* random number in order to resample from the n_p disjoint sets, i.e.

Sample $r \sim \mathcal{U}[0, 1/n_p]$

For $i = 1, \dots, n_p$

If $\sum_{m=1}^{j-1} \tilde{w}_k^{(m)} < r + (i-1)/n_p$ but $\sum_{m=1}^j \tilde{w}_k^{(m)} \geq r + (i-1)/n_p$, set $\vec{x}^{(i)} = \vec{x}^{(j)}$

In practice, any of the residual, stratified, or systematic resampling methods have similar performance, but are all better than the simple multinomial resampling method.

Bootstrap Particle Filter

The **bootstrap particle filter (BPF)** uses the state transition conditional PDF as the importance function and one of the four resampling methods and can be outlined as follows:

First, the **BPF initialization step** samples the initial n_p particles from an initial distribution as

$$\vec{x}_0^{(i)} \sim f_{\vec{X}_0}(\vec{x}_0) \quad i = 1, \dots, n_p \quad (16.209)$$

Then, the recursive **BPF SIR step** performs the sequential importance resampling as

S Sample n_p particles from the importance function.

For $i = 1, \dots, n_p$, sample the state transition PDF with \vec{W}_{k-1} as the random variate

$$\vec{x}_k^{(i)} \sim f_{\vec{X}_k | \vec{X}_{k-1}}(\vec{x}_k | \vec{x}_{k-1|k-1}^{(i)}) = f_{k-1}(\vec{x}_{k-1|k-1}^{(i)}, \vec{u}_{k-1}, \vec{W}_{k-1}) \quad (16.210)$$

I Compute the Importance weights.

For $i = 1, \dots, n_p$, compute the relative importance weights by

$$w_k^{(i)} = f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_{k|k-1}^{(i)}) \quad (16.211)$$

For $i = 1, \dots, n_p$, compute the normalized importance weights by

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{n_p} w_k^{(i)}} \quad (16.212)$$

R Resample $\vec{x}_k^{(i)}$ using the multinomial, residual, stratified, or systematic method with weights, $\tilde{w}_k^{(i)}$

Finally, the **BPF state extraction step** extracts the chosen statistics on the estimated posterior distribution of \vec{X}_k . e.g., the mean

$$\hat{\vec{x}}_{PF} = \frac{1}{n_p} \sum_{i=1}^{n_p} \vec{x}_k^{(i)} \quad (16.213)$$

and covariance

$$\hat{P}_{PF} = \frac{1}{n_p - 1} \sum_{i=1}^{n_p} \left(\vec{x}_k^{(i)} - \hat{\vec{x}}_{PF} \right) \left(\vec{x}_k^{(i)} - \hat{\vec{x}}_{PF} \right)^T \quad (16.214)$$

Particle Depletion and Mitigation

Furthermore, since the resampling step favors the selection of particles with higher weights, particles with low weights may be eliminated altogether after the resampling while others may have many copies made, a phenomenon known as **particle depletion**, also known as **sample impoverishment**. Therefore, resampling methods typically use additional steps to counteract particle depletion.

One simple method is known as **roughening** which adds zero-mean white noise, $n^{(i)}$, typically Gaussian, to each particle after the resampling process. Here, the m^{th} element of the vector $\vec{x}_{k|k}^{(i)}$ with $m = 1, \dots, n_x$ can be modified by an additional sampling procedure

$$\vec{x}_{k|k}^{(i)}(m) = \vec{x}_{k|k}^{(i)} + \sqrt{\alpha M(m) n_p^{-1/n_x}} n^{(i)} \quad (16.215)$$

where α is a tuning parameter, e.g. $\alpha = 0.2$, that specifies the amount of jitter added to each particle, $M(m)$ is the maximum difference between elements of the particle vectors before roughening, i.e.

$$M(m) = \max_{i,j} |\vec{x}_{k|k}^{(i)}(m) - \vec{x}_{k|k}^{(j)}(m)| \quad (16.216)$$

and $n^{(i)} \sim \mathcal{N}(0, I)$.

Another form of roughening on the prior particles generated from the importance function is known as **prior editing**. This involves the rejection and resampling of an prior sample, $\vec{x}_{k|k-1}^{(i)}$, before the resampling procedure if it falls in a region of the state space with a small importance weight $\tilde{w}_k^{(i)}$. If the prior sample is

rejected, then it is roughened as many times as necessary until it is in a region of significant probability, e.g. within $6\sigma_y$ of $\vec{y}_k - h(\vec{x}_{k|k-1}^{(i)})$.

A more robust version of prior editing can be achieved through the use of a **Markov chain Monte Carlo (MCMC) sampling**. This approach moves the prior particle, $\vec{x}_{k|k-1}^{(i)}$, to a new randomly generated state, $\vec{x}_{k|k-1}^{(i)'}$ if a uniformly distributed random number is less than an **acceptance probability**, α . While there are different methods for computing this acceptance probability, it is generally computed as the probability that the prior particle is *consistent* with the measurement, relative to the probability that new randomly generated state is *consistent* with the measurement. In the **Metropolis-Hastings algorithm**, the acceptance probability is given by

$$\alpha = \min \left[1, \frac{f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_{k|k-1}^{(i)'}) f_x(\vec{x}_{k|k-1}^{(i)' | \vec{x}_{k-1|k-1}^{(i)}})}{f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_{k|k-1}^{(i)}) f_x(\vec{x}_{k|k-1}^{(i)} | \vec{x}_{k-1|k-1}^{(i)})} \right] \quad (16.217)$$

where the first fraction is the ratio of the likelihood conditioned on the new particle to the likelihood conditioned on the old particle. The second fraction is the ratio of the probability of the new particle transitioning to the old particle transitioning. The acceptance probability is the product of these two fractions which increases as the probability of the new particle increases. The old prior particle, $\vec{x}_{k|k-1}^{(i)}$, is changed to the new particle, $\vec{x}_{k|k-1}^{(i)'}$, if the old particle has a low probability of being selected with the resampling step, thereby maintaining diversity in the particles that result from the resampling step.

References

For more information, please refer to the following

- Sarkka, S., “7.3 Sequential importance sampling,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 120-123
- Sarkka, S., “7.4 Sequential importance resampling,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 123-129
- Simon, D., “15.2 Particle filtering,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 466-469

16.6 Alternative Particle Filtering and Smoothing

Regularized and Adaptive Resampling

Instead of importance resampling via multinomial, residual, stratified, or systematic methods, an alternative formulation is the **regularized resampling** which resamples from a continuously-valued approximation of the posterior conditional PDF instead of sampling from the PMF approximation, i.e., instead of resampling from

$$\hat{f}_{\vec{X}_k | \vec{Y}_k}(\vec{x}_k | \vec{y}_k) = \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \delta(\vec{x}_k - \vec{x}_k^{(i)}) \quad (16.218)$$

one resamples from

$$\hat{f}_{\vec{X}_k|\vec{Y}_k}(\vec{x}_k|\vec{y}_k) = \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \mathcal{K}_h(\vec{x}_k - \vec{x}_k^{(i)}) \quad (16.219)$$

where $\mathcal{K}_h(\bullet)$ is a scaled kernel PDF, i.e.

$$\mathcal{K}_h(\vec{x}_k - \vec{x}_k^{(i)}) = h^{-n_x} \mathcal{K}\left(\frac{\vec{x}_k - \vec{x}_k^{(i)}}{h}\right) \quad (16.220)$$

where h is the positive scalar kernel bandwidth, and $\mathcal{K}(\bullet)$ is a **kernel PDF** with zero-mean and finite variance.

$K(\bullet)$ and h are typically chosen to minimize a measure of the error between the true posterior PDF and the approximation, i.e.

$$\mathcal{K}_{opt}(\vec{x}), h_{opt} = \underset{\mathcal{K}, h}{\operatorname{argmin}} \int \cdots \int \left(\hat{f}_{\vec{X}_k|\vec{Y}_k}(\vec{x}_k|\vec{y}_k) - f_{\vec{X}_k|\vec{Y}_k}(\vec{x}_k|\vec{y}_k) \right)^2 d\vec{x} \quad (16.221)$$

In the case of equal weights, i.e. $\tilde{w}_k^{(i)} = n_p^{-1}$ for all i , the optimal kernel is the **Epanechnikov kernel**

$$\mathcal{K}_{opt}(\vec{x}) = \begin{cases} \frac{n_x+2}{2V_n} (1 - \|\vec{x}\|_2^2) & \text{if } \|\vec{x}\|_2 < 1 \\ 0 & \text{otherwise} \end{cases} \quad (16.222)$$

where V_n is the volume of the n_x -dimensional unit hypersphere which can be shown for $n_x = 1$, i.e., a line, $V_n = 2$, for $n_x = 2$, i.e., a circle, $V_n = \pi$, for $n_x = 3$, i.e., a sphere, $V_n = 4\pi/3$, and for $n_x \geq 3$, $V_n = 2\pi V_{n-2}/n_x$. Furthermore, if $f_{\vec{X}|\vec{Y}_k}(\vec{x}_k|\vec{y}_k)$ is a multivariate Gaussian with an identity matrix, then the optimal bandwidth is

$$h_{opt} = \left(8V_n^{-1}(n_x + 4)(2\sqrt{\pi})^{n_x} \right)^{1/(n_x+4)} n_p^{-1/(n_x+4)} \quad (16.223)$$

where, to handle multimodal PDF's one should use $h = h_{opt}/2$.

Although these choices are optimal for only equal weights and a Gaussian PDF, they are often used in practice and the bandwidth, h , is often treated as a tuning parameter for the regularized resampling. Note that the regularization produces a continuous approximation as a weighted sum of these kernel PDFs with the normalized weights. Furthermore, since this regularization procedure assumes $f_{\vec{X}|\vec{Y}_k}(\vec{x}_k|\vec{y}_k)$ has unity variance, one should numerically compute the covariance of $\vec{x}_k^{(i)}$ at each time step, i.e., P_k . Then, as a kernel, one should use

$$\mathcal{K}_h(\vec{x}) = \left(\det P_k^{1/2} \right)^{-1} h^{-n_x} \mathcal{K}_{opt}\left(\frac{P_k^{-1/2} \vec{x}}{h}\right) \quad (16.224)$$

As the resampling step of particle filtering can be computationally expensive, often one can employ **adaptive resampling** which monitors the “effective” number of particles in the SIS approach at time step k , $\hat{n}_{p,k}^{eff}$, which can be estimated from the variance of the importance weights, i.e.,

$$\hat{n}_{p,k}^{eff} = \frac{1}{\sum_{i=1}^{n_p} (\tilde{w}_k^{(i)})^2} \quad (16.225)$$

and implements a resampling step at time step k if

$$\frac{\hat{n}_{p,k}^{eff}}{n_p} < \tau_{eff} \quad (16.226)$$

where τ_{eff} is the **effective number threshold** for the test statistic, $\hat{n}_{p,k}^{eff}/n_p$, e.g., 0.1.

When implementing adaptive resampling, one must recursively update the importance weights for $i = 1, \dots, n_p$ as

$$w_k^{(i)} = \frac{f_{\vec{Y}_k | \vec{X}_k}(\vec{y}_k | \vec{x}_k^{(i)}) f_{\vec{X}_k}(\vec{x}_k^{(i)} | \vec{x}_{k-1}^{(i)})}{\pi(\vec{x}_k^{(i)} | \vec{x}_{0:k-1}^{(i)}, \vec{y}_{1:k})} \tilde{w}_{k-1}^{(i)} \quad (16.227)$$

and then for $i = 1, \dots, n_p$, compute the normalized importance weights by

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{n_p} w_k^{(i)}} \quad (16.228)$$

After a resampling step is performed, one resets the normalized importance weights to

$$\tilde{w}_k^{(i)} = \frac{1}{n_p} \quad (16.229)$$

which can be used in the recursion for the importance weights.

Alternative Importance Function Approximation as Gaussian

Instead of using the state transition PDF as the importance function, i.e.,

$$\pi(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k}) = f_X(\vec{x}_k | \vec{x}_{k-1}) \quad (16.230)$$

an alternative for a particle filter is to update each prior particle according to a multivariate Gaussian approximation of the optimal importance function, i.e.,

$$\pi(\vec{x}_k | \vec{x}_{0:k-1}, \vec{y}_{1:k}) = f_N(\vec{x}_k; \hat{\vec{x}}_{k|k-1}, P_{k|k-1}) \quad (16.231)$$

Here, for each *single* particle, $i = 1, \dots, n_p$

$$\vec{X}_{k|k-1}^{(i)} \sim N(\hat{\vec{x}}_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) \quad (16.232)$$

where the hyper-parameters of each multivariate Gaussian importance function approximation is updated via an EKF or one of the SPKFs. This alternative method is known as an **extended Kalman particle filter (EKF)** or a **sigma-point particle filter (SPPF)**, respectively. This use of an additional filter for *each* particle directly improves the importance function approximation and improves the PF performance in many cases, albeit at significant computational and memory costs. The **extended Kalman/sigma-point particle filter (EK/SPPF)** uses n_p EKF/SPKFs to approximate the importance function for each particle as a multivariate Gaussian and can be outlined as follows.

First, the **EKF/SPPF initialization step** samples the initial $i = 1, \dots, n_p$ particles from an initial distribution for the state, importance weights, and state covariances as

$$\begin{aligned}\hat{x}_{0|0}^{(i)} &\sim f_{\vec{X}_0}(\vec{x}_0) \\ w_0^{(i)} &= f_{\vec{X}_0}(\vec{x}_{0|0}^{(i)}) \\ \tilde{w}_0^{(i)} &= \frac{w_0^{(i)}}{\sum_{i=1}^{n_p} w_0^{(i)}} \\ \text{Assign } P_{0|0}^{(i)}\end{aligned}\tag{16.233}$$

Also, if implementing adaptive resampling, one must choose an effective number threshold ratio for resampling, τ_{eff} .

Then, the recursive **EKF/SPPF SIR step** performs the sequential importance resampling as

S Sample particles from the importance functions.

For $i = 1, \dots, n_p$,

P Compute $\vec{x}_{k|k-1}^{(i)}$ and $P_{k|k-1}^{(i)}$ with the prediction step of the corresponding EKF/SPKF

C Compute $\vec{x}_{k|k}^{(i)}$ and $P_{k|k}^{(i)}$ with the correction step of the corresponding EKF/SPKF

Sample $\vec{x}_k^{(i)} \sim \mathcal{N}(\vec{x}_{k|k}^{(i)}, P_{k|k}^{(i)})$ as the i^{th} importance function from the EKF/SPKF

I Compute the importance weights.

For $i = 1, \dots, n_p$, compute the relative importance weights by

$$w_k^{(i)} = \frac{f_{\vec{Y}_k|\vec{X}_k}(\vec{y}_k | \vec{x}_{k|k-1}^{(i)}) f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}_k^{(i)} | \vec{x}_{k-1|k-1}^{(i)})}{f_N(\vec{x}_k^{(i)}; \vec{x}_{k|k}, P_{k|k}^{(i)})} \tilde{w}_{k-1}^{(i)}\tag{16.234}$$

For $i = 1, \dots, n_p$, compute the normalized importance weights by

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{n_p} w_k^{(i)}}\tag{16.235}$$

R Resample $\vec{x}_k^{(i)}$ with weights, $\tilde{w}_k^{(i)}$, if

$$\frac{1}{n_p \sum_{i=1}^{n_p} (\tilde{w}_k^{(i)})^2} < \tau_{eff}\tag{16.236}$$

Reset the relative importance weights as

$$\tilde{w}_k^{(i)} = \frac{1}{n_p} \quad (16.237)$$

Finally, the **EK/SPPF state extraction step** extracts the chosen statistics on the estimated posterior distribution of \vec{x}_k . e.g., the state mean and covariance as

$$\begin{aligned}\hat{\vec{x}}_{k,PF} &= \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \vec{x}_k^{(i)} \\ \hat{P}_{k,PF} &= \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \left(\vec{x}_k^{(i)} - \hat{\vec{x}}_{k,PF} \right) \left(\vec{x}_k^{(i)} - \hat{\vec{x}}_{k,PF} \right)^T\end{aligned}\quad (16.238)$$

Rao-Blackwellized Particle Filter

The Rao-Blackwellized particle filter provides a Bayesian framework to evaluate some of the filtering equations analytically and the others with sequential Monte Carlo sampling. This approach is preferred, not just for the computational savings, but the less variance that can be proven via the **Rao-Blackwell theorem**. The Rao-Blackwellized particle filter considers marginalized filtering of the state, \vec{x}_k , which is a conditionally linear-Gaussian, or an approximate linear-Gaussian, HMM if one knows the **latent variables** in vectorized form, $\vec{\phi}_k$, i.e.,

$$\begin{aligned}f_{\vec{X}_k | \vec{X}_{k-1}, \vec{\Phi}_{k-1}}(\vec{x}_k | \vec{x}_{k-1}, \vec{\phi}_{k-1}) &= f_N\left(\vec{x}_k; f_{k-1}(\vec{x}_{k-1}, \vec{\phi}_k, \vec{u}_{k-1}), Q_{k-1}(\vec{\phi}_{k-1})\right) \\ f_{\vec{Y}_k | \vec{X}_k, \vec{\Phi}_k}(\vec{y}_k | \vec{x}_k, \vec{\phi}_k) &= f_N\left(\vec{y}_k; h_k(\vec{x}_k, \vec{\phi}_k), R_k(\vec{\phi}_k)\right) \\ f_{\vec{\Phi}_k | \vec{\Phi}_{k-1}}(\vec{\phi}_k | \vec{\phi}_{k-1}) &= \text{Markov model}\end{aligned}\quad (16.239)$$

where \vec{y}_k is the corrective measurement and \vec{u}_{k-1} is the input or predictive measurement.

This corresponds to an approximation of the joint filtering distribution as

$$f_{\vec{X}_k, \vec{\Phi}_k | \vec{Y}_{1:k}}(\vec{x}_k, \vec{\phi}_k | \vec{y}_{1:k}) \approx \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \delta(\vec{\phi}_k - \vec{\phi}_k^{(i)}) f_N(\vec{x}_k; \hat{\vec{x}}_k^{(i)}, P_k^{(i)}) \quad (16.240)$$

As the joint posterior distribution at time step k , one has

$$f_{\vec{X}_{0:k}, \vec{\Phi}_{0:k} | \vec{Y}_{1:k}}(\vec{x}_{0:k}, \vec{\phi}_{0:k} | \vec{y}_{1:k}) = f_{\vec{X}_{0:k} | \vec{\Phi}_{0:k}, \vec{Y}_{1:k}}(\vec{x}_{0:k} | \vec{\phi}_{0:k}, \vec{y}_{1:k}) f_{\vec{\Phi}_{0:k} | \vec{Y}_{1:k}}(\vec{\phi}_{0:k} | \vec{y}_{1:k}) \quad (16.241)$$

where the first term on the right side can be recursively updated using a linear or nonlinear Kalman filter with a fixed trajectory $\vec{\Phi}_{0:k} = \vec{\phi}_{0:k}$ while the second term on the right side can be recursively updated according to Bayes rule, i.e.,

$$f_{\vec{\Phi}_{0:k} | \vec{Y}_{1:k}}(\vec{\phi}_{0:k} | \vec{y}_{1:k}) = \frac{f_{\vec{Y}_k | \vec{\Phi}_{0:k}, \vec{Y}_{1:k}}(\vec{y}_k | \vec{\phi}_{0:k}, \vec{y}_{1:k-1}) f_{\vec{\Phi}_{0:k} | \vec{Y}_{1:k-1}}(\vec{\phi}_{0:k} | \vec{y}_{1:k-1})}{f_{\vec{Y}_k}(\vec{y}_k)} \quad (16.242)$$

which by the Markov property for $\vec{\phi}_k$, one has

$$f_{\vec{\Phi}_{0:k} | \vec{Y}_{1:k}}(\vec{\phi}_{0:k} | \vec{y}_{1:k}) = \frac{f_{\vec{Y}_k | \vec{\Phi}_{0:k}, \vec{Y}_{1:k}}(\vec{y}_k | \vec{\phi}_{0:k}, \vec{y}_{1:k-1}) f_{\vec{\Phi}_k | \vec{\Phi}_{k-1}}(\vec{\phi}_k | \vec{\phi}_{k-1}) f_{\vec{\Phi}_{0:k-1} | \vec{Y}_{1:k-1}}(\vec{\phi}_{0:k-1} | \vec{y}_{1:k-1})}{f_{\vec{Y}_k}(\vec{y}_k)} \quad (16.243)$$

which can be updated via a particle filter recursion if one chooses an importance function at time step k with a recursion

$$\pi(\vec{\phi}_{0:k} | \vec{y}_{1:k}) = \pi(\vec{\phi}_k | \vec{\phi}_{0:k-1}, \vec{y}_{1:k}) \pi(\vec{\phi}_{0:k-1} | \vec{y}_{1:k-1}) \quad (16.244)$$

and corresponding relative importance weight recursion for $i = 1, \dots, n_p$ particles as

$$w_k^{(i)} = \frac{f_{\vec{Y}_k | \vec{\Phi}_{0:k}, \vec{Y}_{1:k}}(\vec{y}_k | \vec{\phi}_{0:k}^{(i)}, \vec{y}_{1:k-1}) f_{\vec{\Phi}_k | \vec{\Phi}_{k-1}}(\vec{\phi}_k^{(i)} | \vec{\phi}_{k-1}^{(i)})}{\pi(\vec{\phi}_k^{(i)} | \vec{\phi}_{0:k-1}^{(i)}, \vec{y}_{1:k})} \tilde{w}_k^{(i)} \quad (16.245)$$

The **Rao-Blackwellized particle filter (RBPF)**, also known as the **marginalized particle filter (MPF)** can outlined as follows. First, the **RBPF initialization step** samples the initial $i = 1, \dots, n_p$ particles from an initial distribution as the state estimates, latent variable vector, importance weights, and

$$\begin{aligned} \hat{x}_{0|0}^{(i)} &\sim f_{\vec{X}_0}(\vec{x}_0) \\ \vec{\phi}_{0|0}^{(i)} &\sim f_{\vec{\Phi}_0}(\vec{\phi}_0) \\ w_0^{(i)} &= f_{\vec{X}_0}(\vec{x}_{0|0}^{(i)}) \\ \tilde{w}_0^{(i)} &= \frac{w_0^{(i)}}{\sum_{i=1}^{n_p} w_0^{(i)}} \\ \text{Assign } P_{0|0}^{(i)} & \end{aligned} \quad (16.246)$$

Then, the recursive **RBPF KF-SIR step** performs the prediction, sampling, importance computation, correction, and resampling steps as

P Perform (nonlinear) Kalman filter prediction steps for $i = 1, \dots, n_p$ prior state estimates, $\vec{x}_{k|k-1}^{(i)}$, and state covariances, $P_{k|k-1}^{(i)}$ with the process model conditioned on $\vec{\phi}_{k-1}^{(i)}$

S Sample n_p particles by

For $i = 1, \dots, n_p$

$$\vec{\phi}_k^{(i)} \sim \pi(\vec{\phi}_k | \vec{\phi}_{0:k-1}^{(i)}, \vec{y}_{1:k}) \quad (16.247)$$

I Compute the importance weights:

For $i = 1, \dots, n_p$

$$w_k^{(i)} = \frac{f_{\vec{Y}_k | \vec{\Phi}_{0:k}, \vec{Y}_{1:k}}(\vec{y}_k | \vec{\phi}_{0:k}^{(i)}, \vec{y}_{1:k-1}) f_{\vec{\Phi}_k | \vec{\Phi}_{k-1}}(\vec{\phi}_k^{(i)} | \vec{\phi}_{k-1}^{(i)})}{\pi(\vec{\phi}_k^{(i)} | \vec{\phi}_{0:k-1}^{(i)}, \vec{y}_{1:k})} \tilde{w}_k^{(i)} \quad (16.248)$$

- where $f_{\vec{Y}_k | \vec{\Phi}_{0:k}, \vec{Y}_{1:k}}(\vec{y}_k | \vec{\phi}_{0:k}^{(i)}, \vec{y}_{1:k-1})$ is the marginal measurement likelihood based on the Gaussian approximation with $\hat{\vec{x}}_{k|k-1}$ and $P_{k|k-1}^{(i)}$.

and normalize the importance weights by:

$$\tilde{w}_k^{(i)} = \frac{w_k^{(i)}}{\sum_{i=1}^{n_p} w_k^{(i)}} \quad (16.249)$$

C Perform (nonlinear) Kalman filter correction steps for $i = 1, \dots, n_p$ posterior state estimates, $\vec{x}_{k|k}^{(i)}$, and state covariances, $P_{k|k}^{(i)}$ with the measurement model conditioned on $\vec{\phi}_k^{(i)}$

R Resample $\vec{\phi}_k^{(i)}$ with weights, $\tilde{w}_k^{(i)}$ if

$$\frac{1}{n_p \sum_{i=1}^{n_p} (\tilde{w}_k^{(i)})^2} < \tau_{eff} \quad (16.250)$$

Reset the relative importance weights as

$$\tilde{w}_k^{(i)} = \frac{1}{n_p} \quad (16.251)$$

Finally, the **RBPF state extraction step** extracts the chosen statistics on the estimated joint posterior distribution of $\vec{X}_k, \vec{\Phi}_k$. e.g., the state mean, state covariance, latent variables mean, and latent variables covariance as

$$\begin{aligned} \hat{\vec{x}}_{k,PF} &= \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \vec{x}_k^{(i)} \\ P_{k,PF} &= \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} P_k^{(i)} \\ \hat{\vec{\phi}}_{k,PF} &= \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \vec{\phi}_k^{(i)} \\ \hat{P}_{k,PF} &= \sum_{i=1}^{n_p} \tilde{w}_k^{(i)} \left(\vec{\phi}_k^{(i)} - \hat{\vec{\phi}}_{k,PF} \right) \left(\vec{\phi}_k^{(i)} - \hat{\vec{\phi}}_{k,PF} \right)^T \end{aligned} \quad (16.252)$$

Particle Smoothers

While the same SIR method can provide the state smoothing distribution by sequentially updating the importance weights and resampling the entire state trajectories instead of simply the current states. However, such an approach typically results in a degenerate distribution for the k^{th} time step when $k \ll N$. Thus, two alternative particle smoothers are summarized here.

The **backward-simulation particle smoother (BS-PS)** updates the state filtering distributions approximated as a weighted set of particles, $w_k^{(i)}$ and $\vec{x}_k^{(i)}$ for $i = 1, \dots, n_p$ and $k = 1, \dots, N$. The initialization step by resampling $\vec{x}_N^{(i)}$ as $\tilde{x}_N^{(i)}$. Then, one recursively approximates the state smoothing distribution for $k = N - 1, \dots, 0$ by first computing the relative importance weights as

$$w_{k|k+1}^{(i)} = f_{\vec{X}_{k+1}|\vec{X}_k}(\tilde{x}_{k+1}^{(i)}|\vec{x}_k^{(i)})w_k^{(i)} \quad (16.253)$$

and the normalized importance weights as

$$\tilde{w}_{k|k+1}^{(i)} = \frac{w_{k|k+1}^{(i)}}{\sum_{i=1}^{n_p} w_{k|k+1}^{(i)}} \quad (16.254)$$

and resamples $\tilde{x}_k^{(i)}$ with weights $\tilde{w}_{k|k+1}^{(i)}$.

The **reweighted particle smoother (RPS)** is implemented by recomputing the normalized importance weights for $k = N - 1, \dots, 0$

$$\tilde{w}_{k|N}^{(i)} = \sum_{j=1}^{n_p} \tilde{w}_{k+1|N}^{(j)} \frac{\tilde{w}_k^{(i)} f_{\vec{X}_{k+1}|\vec{X}_k}(\vec{x}_{k+1}^{(j)}|\vec{x}_k^{(i)})}{\sum_{\ell=1}^{n_p} \tilde{w}_{k+1|N}^{(\ell)} f_{\vec{X}_{k+1}|\vec{X}_k}(\vec{x}_{k+1}^{(\ell)}|\vec{x}_k^{(i)})} \quad (16.255)$$

where one defines $\tilde{w}_{N|N}^{(i)} = \tilde{w}_N^{(i)}$ for $i = 1, \dots, n_p$

References

For more information, please refer to the following

- Sarkka, S., “7.4 Sequential importance resampling,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 123-129
- Sarkka, S., “7.5 Rao-Blackwellized particle filter,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 129-132
- Sarkka, S., “11.1 SIR particle smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 165-167
- Sarkka, S., “11.2 Backward-simulation particle smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 165-167
- Sarkka, S., “11.3 SIR particle smoother,” in *Bayesian Filtering and Smoothing*, Cambridge University Press, 2013, pp. 165-167
- Simon, D., “15.3 Implementation Issues,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 469-480

Advanced Kalman Filtering Theory

17.1 Error-State Kalman Filtering

Error-State Kalman Filtering

In **error-state Kalman filtering** at each time step k , one considers the **true-state vector**, \vec{x} , as composed of the **nominal-state vector**, \bar{x} , and the **error-state vector**, i.e.

$$\vec{x} = \bar{x} \oplus \delta \vec{x} \quad (17.1)$$

where \oplus is any general composition where each state vector can be written as some continuous-time stochastic state-space model

$$\begin{aligned} \dot{\vec{x}} &= f(\vec{x}, \vec{u}, \vec{w}) \\ \vec{y} &= h(\vec{x}, \vec{v}) \end{aligned} \quad (17.2)$$

where, because of the potential composition of the state vector, the state dynamics for the nominal-state is

$$\dot{\bar{x}} = \bar{f}(\bar{x}, \vec{u}, \vec{w}) \quad (17.3)$$

and the state dynamics for the error-state is

$$\delta \dot{\vec{x}} = f_\delta(\bar{x}, \delta \vec{x}, \vec{u}, \vec{w}) \quad (17.4)$$

where \vec{u} is the proprioceptive sensor measurement or control input, \vec{y} is the exteroceptive sensor measurement, \vec{W} is a zero-mean stationary noise process with power spectral density, S , and \vec{W} is a zero-mean stationary noise process with covariance, R .

In this case, if there is a non-zero error-state estimate, one should perform a **state update step** by updating the nominal-state estimate, \hat{x} , with the error-state estimate, $\delta \hat{\vec{x}}$, i.e.

$$\hat{x}' = \hat{x} \oplus \delta \hat{\vec{x}} \quad (17.5)$$

where $\hat{\vec{x}}'$ is the updated nominal-state estimate. Then, the error-state is reset via the relationship

$$\delta\vec{x}' = g(\delta\vec{x}) = \delta\vec{x} \ominus \hat{\delta}\vec{x} \quad (17.6)$$

where $\delta\vec{x}'$ is the updated error-state estimate and \ominus stands for the composition inverse of \oplus . Thus, the error-state estimate is updated to zero

$$\hat{\delta}\vec{x}' = \vec{0} \quad (17.7)$$

and the error-state covariance, P , is updated as

$$P' = \left(\frac{\partial g}{\partial \delta\vec{x}} \right)_{\delta\vec{x}=\hat{\delta}\vec{x}} P \left(\frac{\partial g}{\partial \delta\vec{x}} \right)_{\delta\vec{x}=\hat{\delta}\vec{x}}^T \quad (17.8)$$

From control inputs or proprioceptive sensor measurements at time steps $k - 1$ and k , the **error-state prediction step**, also known as the **error-state time update step**, estimates the nominal-state, $\hat{\vec{x}}$ using an integration of the dynamics equation, i.e., $\bar{f}()$. The error-state estimate remains zero as any constant *known* bias should be encompassed in the nominal-state dynamics. However, the covariance of the error-state must be estimated from time step $k - 1$ to k for use with the **error-state correction step**, also known as the **error-state measurement update step**, using a Kalman filter framework.

Error-State Extended Kalman Filter

For the **error-state extended Kalman filter (ES-EKF)**, one can use a continuous-time update for the nominal-state estimate as

$$\hat{\vec{x}}_{k|k-1} = \hat{\vec{x}}_{k-1|k-1} + \int_{t(k-1)}^{t(k)} \bar{f}(\hat{\vec{x}}, \vec{u}, 0) \quad (17.9)$$

or a discrete-time update for the nominal-state estimate as

$$\hat{\vec{x}}_{k|k-1} = \bar{f}(\hat{\vec{x}}_{k-1|k-1}, \vec{u}_{k-1}, 0) \quad (17.10)$$

However, for the error-state covariance update, one uses the linearized state matrix as

$$A_{k-1} = \left[\frac{\partial f_\delta}{\partial \delta\vec{x}} \right]_{\bar{x}=\hat{\vec{x}}_k, \delta\vec{x}=0, \vec{u}=\vec{u}_{k-1}, \vec{w}=0} \quad (17.11)$$

and the linearized process noise gain matrix

$$L_{k-1} = \left[\frac{\partial f_\delta}{\partial \vec{w}} \right]_{\bar{x}=\hat{\vec{x}}_k, \delta\vec{x}=\vec{0}, \vec{u}=\vec{u}_{k-1}, \vec{w}=0} \quad (17.12)$$

to form the discretized state matrix with sampling time interval, Δt , from time step $k - 1$ to k , as

$$F_{k-1} = \exp(A_{k-1}\Delta t) \quad (17.13)$$

and discretized noise covariance Q_{k-1} from the PSD, S , of the process noise \vec{W} as

$$Q_{k-1} = \int_0^{\Delta t} \exp(A_{k-1}\tau) L_{k-1} S L_{k-1}^T \exp(A_{k-1}\tau)^T d\tau \quad (17.14)$$

It should be noted that Q_{k-1} can be efficiently computed using the following representation

$$\begin{bmatrix} E_{11} & E_{12} \\ E_{21} & E_{22} \end{bmatrix} = \exp \left(\begin{bmatrix} -A_{k-1} & L_{k-1} S L_{k-1}^T \\ 0 & A_{k-1}^T \end{bmatrix} \Delta t \right) \quad (17.15)$$

and then multiplying

$$Q_{k-1} = E_{22}^T E_{12} \quad (17.16)$$

where notably, a first-order approximation of matrix exponential would provide

$$Q_{k-1} = (I + \Delta t A_{k-1}) \left(\Delta t L_{k-1} S L_{k-1}^T \right) \quad (17.17)$$

Finally, one can form the prior error-state covariance as

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + Q_{k-1} \quad (17.18)$$

At time step k , the **error-state measurement update step** compares an exteroceptive sensor measurement, \vec{y}_k , with the expected measurement based on the nominal-state, i.e.

$$\hat{\vec{y}}_k = h(\hat{\vec{x}}_{k|k-1}, \vec{v}_k) \quad (17.19)$$

where \vec{v}_k is white measurement noise with covariance R_k . For the ES-EKF, one computes the innovation as

$$\delta \vec{y}_k = \vec{y}_k - \hat{\vec{y}}_k \quad (17.20)$$

the linearized output matrix as

$$H_k = \left[\frac{\partial h}{\partial \vec{x}} \right]_{\vec{x}=\hat{\vec{x}}_k, \vec{v}=0} \quad (17.21)$$

and the linearized measurement noise gain matrix

$$M_k = \left[\frac{\partial h}{\partial \vec{v}} \right]_{\vec{x}=\hat{\vec{x}}_k, \vec{v}=0} \quad (17.22)$$

to form the approximate innovation covariance noise covariance

$$S_k = H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \quad (17.23)$$

and the sub-optimal Kalman gain as

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (17.24)$$

Finally, one can form the posterior error-state mean estimate as

$$\hat{\vec{x}}_k = K_k \delta \vec{y}_k \quad (17.25)$$

and the approximate posterior error-state covariance as

$$P_{k|k} = P_{k|k-1} - K_k S_k K_k^T \quad (17.26)$$

The next step in the recursion uses this error-state estimated to update the nominal-state in the state update step.

Lastly, it should be noted that the **multiplicative extended Kalman filter (MEKF)** is a special case of the ES-EKF when the true-state composition between the nominal-state and the error-state is multiplicative, e.g., in attitude state estimation.

Error-State Sigma-Point Kalman Filtering

References

For more information, please refer to the following

- Barrau, A., and Bonnabel, S., “The invariant extended Kalman filter as a stable observer,” in *IEEE Transactions on Automatic Control*, 62(4), 2017, pp. 1797–1812
- Madyastha, V., Ravindra, V., Srinath Mallikarjunan, S., and Goyal, A., “Extended Kalman Filter vs. Error State Kalman Filter for Aircraft Attitude Estimation,” in *AIAA Guidance, Navigation, and Control Conference*, 2011
- Simon, D., “13.3 Higher-order approaches,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 410-420

17.2 Multi-Modal and Heavy-Tailed Kalman Filtering

Gaussian Sum Filter

The use of nonlinear Kalman filters for non-Gaussian distributions can also be approximated by the fact that any distribution can be approximated arbitrarily close in the L_1 -sense as a Gaussian mixture (GM) distribution, or a weighted “sum” of Gaussians. The so-called **Gaussian sum filter (GSF)** approximates the true distributions of the process noise and measurement noise as a GM of n_N elements with n_N weights, means, and covariances, runs n_N parallel nonlinear Kalman filters for each element in the , and combine them for an approximately optimal estimate. Choosing n_N is a trade-off between approximation accuracy/optimality and computational cost. It should also be mentioned that for noise with heavy tails, the Gaussian

Kalman-Levy Filtering

References

For more information, please refer to the following

- Simon, D., “13.3 Higher-order approaches,” in *Optimal State Estimation*, 1st Ed., John Wiley & Sons, 2006, pp. 410-420

17.3 Variational Bayes Kalman Filtering

Recall that in the previous discrete-time Kalman filter recursion, one typically assumes a linear stochastic state-space model as

$$\begin{aligned}\vec{x}_k &= F_{k-1} \vec{x}_{k-1} + G_{k-1} \vec{u}_{k-1} + \vec{w}_{k-1} \\ \vec{y}_k &= H_k \vec{x}_k + \vec{v}_k\end{aligned}\tag{17.27}$$

with white noise $\vec{W}_k \sim \mathcal{N}(0, Q_k)$ and $\vec{V}_k \sim \mathcal{N}(0, R_k)$ which are uncorrelated with each other.

Joint State and Noise Parameter Kalman Filter

Variational Bayes Adaptive Kalman Filter

GSM Filter

17.4 Distributed Kalman Filtering

Part IV

Aerospace Vehicle Perception Systems

Aerospace Sensor Systems

18.1 Inertial Sensors

Inertial sensors are accelerometers and gyroscopes which are described in terms of the sensing concepts in this section.

Accelerometer

An **accelerometer** is a sensor that directly measures the acceleration of a body in its own instantaneous rest frame. Conceptually, an accelerometer is a **proof mass**, also known as a **seismic mass**, on a spring. For planetary-based accelerometers, the rest frame is the Earth so the gravity force is not included. When the accelerometer experiences an acceleration, the proof mass is deflected to the point that the spring accelerates the mass at the same speed as the casing. The deflection of the proof mass from its neutral position or the compression of the spring provides an acceleration measurement. However, the mass-spring system is also damped so that additional oscillations of the system do not affect the measurement output. Thus, all accelerometers have a frequency response and are designed to provide the required sensitivity and maximum expected acceleration. There are many different designs for converting this mass deflection or spring compression into a measurable electric signal including piezoelectric, piezoresistive, capacitive, and thermal.

Piezoelectric accelerometers use single crystals, e.g. quartz, or piezoceramics, e.g. lead-zirconate-titanate. They are constructed with two piezoelectric transducers. The unit consists of a hollow tube that is sealed by a piezoelectric transducer on each end. The transducers are oppositely polarized and are selected to have a specific series capacitance. The tube is then partially filled with a heavy liquid and the accelerometer is excited. While excited the total output voltage is continuously measured and the volume of the heavy liquid is adjusted until the desired output voltage is obtained. Finally the outputs of the individual transducers are measured, the residual voltage difference is tabulated, and the dominate transducer is identified. They are unmatched in high frequency measurements, low packaged weight, and resistance to high temperatures.

Piezoresistive accelerometers use a semiconductor layer that is attached to a handle wafer by a thick oxide layer. The semiconductor layer is then patterned to the accelerometer's geometry. This semiconductor layer has one or more apertures so that the underlying mass will have the corresponding apertures. Next the semiconductor layer is used as a mask to etch out a cavity in the underlying thick oxide. A mass in the cavity is supported in cantilever fashion by the piezoresistant arms of the semiconductor layer. Directly below the accelerometer's geometry is a flex cavity that allows the mass in the cavity to flex or move in direction that is orthogonal to the surface of the accelerometer. These resist shock better and are able to measure very high accelerations. Integrating piezoresistors in the springs to detect spring deformation, and thus deflection, is a good alternative, although a few more process steps are needed during the fabrication sequence.

Capacitive accelerometer use the properties of an opposed plate capacitor for which the distance between the plates varies proportionally to applied acceleration, thus altering capacitance. This variable is used in a circuit to ultimately deliver a voltage signal that is proportional to acceleration. Capacitive accelerometers are capable of measuring constant as well as slow transient and periodic acceleration. AC capacitive-acceleration sensors fundamentally contain at least two components; the primary is a 'stationary' plate (i.e., connected to the housing) and the secondary plate is attached to the inertial mass, which is free to move inside the housing. These plates form a capacitor whose value is a function of a distance between the plates. The sensing material is either a flat plate of nickel or electronic chip supported above the substrate surface by two torsion bars attached to a central pedestal. This design is simple, reliable, and inexpensive. They measure low frequencies well.

Thermal accelerometer, also known as **convective accelerometers**, contain a small heater in a very small dome. This heats the air or other fluid inside the dome. The thermal bubble acts as the proof mass. An accompanying temperature sensor (like a thermistor; or thermopile) in the dome measures the temperature in one location of the dome. This measures the location of the heated bubble within the dome. When the dome is accelerated, the colder, higher density fluid pushes the heated bubble. The measured temperature changes. The temperature measurement is interpreted as acceleration. The fluid provides the damping. Gravity acting on the fluid provides the spring. Since the proof mass is very lightweight gas, and not held by a beam or lever, thermal accelerometers can survive high shocks. Another variation uses a wire to both heat the gas and detect the change in temperature. The change of temperature changes the resistance of the wire. A two dimensional accelerometer can be economically constructed with one dome, one bubble and two measurement devices.

Gyroscopes

A **gyroscope** is a sensor that directly measures the rotational rate about an axis. Conceptually, it is a spinning wheel or disc in which the spin axis is free to assume any orientation by itself. When rotating, the orientation of this axis is unaffected by tilting or rotation of the mounting, according to the conservation of angular momentum. There are two broad design categories for gyroscopes: gyrostabilized and strapdown. **Gyrostabilized** measure the angles of gimbal bearings or fluid bearings. **Strapdown** measure either optical effects or vibrating structures effects to infer the rotational rate.

A **gimbal** is a ring that is suspended so it can rotate about an axis and nested gimbals with bearings nominally at right angles to each other. Due to the suspended nature of these platforms, the sensors will keep the same orientation relative to the inertial frame while the housing rotates. In addition, two gyroscopes are typically placed on each ring to cancel **gyroscopic precession**, i.e. the tendency of a gyroscope to twist at

right angles to an input torque, thereby changing its axis of rotation. By placing a sensor at the bearings of the gimbals, one can measure the vehicle's attitude by the three sequential rotations of the nested gimbals, i.e. the inertial-to-body-frame Euler angles.

The three major disadvantages to gyrostabilized IMUs are the degradation of the many expensive precision mechanical parts, the mechanical parts' potential to jam, and the fatal phenomenon called gimbal lock. **Gimbal lock** is the loss of one degree of freedom that occurs when the axes of two of the three gimbal rings are driven into a parallel configuration. Because of the parallel orientation of two of the gimbals' axes there is no gimbal available to accommodate rotation about the orthogonal direction to the two axes, thereby "locking" the gyroscope into only measuring rotation in the resulting two-dimensional space.

Consider the direction cosine matrix relationship to the Euler angles

$$C_{G \leftarrow F} = \begin{bmatrix} \cos \theta_y \cos \theta_z & \cos \theta_y \sin \theta_z & -\sin \theta_y \\ \sin \theta_x \sin \theta_y \cos \theta_z - \cos \theta_x \sin \theta_z & \sin \theta_x \sin \theta_y \sin \theta_z + \cos \theta_x \cos \theta_z & \sin \theta_x \cos \theta_y \\ \cos \theta_x \sin \theta_y \cos \theta_z + \sin \theta_x \sin \theta_z & \cos \theta_x \sin \theta_y \sin \theta_z - \sin \theta_x \cos \theta_z & \cos \theta_x \cos \theta_y \end{bmatrix} \quad (18.1)$$

If $\theta_y = \pi/2$, one has

$$C_{G \leftarrow F} = \begin{bmatrix} 0 & 0 & -1 \\ \sin \theta_x \cos \theta_z - \cos \theta_x \sin \theta_z & \sin \theta_x \sin \theta_z + \cos \theta_x \cos \theta_z & 0 \\ \cos \theta_x \cos \theta_z + \sin \theta_x \sin \theta_z & \cos \theta_x \sin \theta_z - \sin \theta_x \cos \theta_z & 0 \end{bmatrix} \quad (18.2)$$

which by trigonometric summation formulas can be rewritten as

$$C_{G \leftarrow F} = \begin{bmatrix} 0 & 0 & -1 \\ \sin(\theta_x - \theta_z) & \cos(\theta_x - \theta_z) & 0 \\ \cos(\theta_x - \theta_z) & -\sin(\theta_x - \theta_z) & 0 \end{bmatrix} \quad (18.3)$$

Similarly, if $\theta_y = -\pi/2$, one has

$$C_{G \leftarrow F} = \begin{bmatrix} 0 & 0 & 1 \\ -\sin \theta_x \cos \theta_z - \cos \theta_x \sin \theta_z & -\sin \theta_x \sin \theta_z + \cos \theta_x \cos \theta_z & 0 \\ -\cos \theta_x \cos \theta_z + \sin \theta_x \sin \theta_z & -\cos \theta_x \sin \theta_z - \sin \theta_x \cos \theta_z & 0 \end{bmatrix} \quad (18.4)$$

which by the trigonometric summation formulas can be rewritten as

$$C_{G \leftarrow F} = \begin{bmatrix} 0 & 0 & 1 \\ -\sin(\theta_x + \theta_z) & \cos(\theta_x + \theta_z) & 0 \\ -\cos(\theta_x + \theta_z) & -\sin(\theta_x + \theta_z) & 0 \end{bmatrix} \quad (18.5)$$

Thus, at the critical angles, $\theta_y = \pm\pi/2$, there is a loss in gimbal bearing angles to distinguish difference between θ_x and θ_z gimbal angles which reduces gyroscope to 2-D, i.e. θ_y and $\theta_x + \theta_z$ or $\theta_x - \theta_z$. In practice, mechanical gimbals encounter difficulties *near* these two critical angles where small rotations of the platform would require large motions of the surrounding gimbals which makes them resist motion due to their inertia, bearing friction, and fluid resistance.

Optical gyroscopes use the **Sagnac effect** of light interferometry to sense the angular motion. Optical gyroscopes are either ring laser or fiber optic gyroscopes. A **ring laser gyroscope (RLG)** uses a single

closed-path with a laser beam while a **fiber optic gyroscope (FOG)** uses multiple coils of the closed-path for a fiber optic cable of known length. A FOG is less sensitive than an RLG to shock and vibration and offers excellent thermal stability, but can be susceptible to magnetic interference on the fiber optics.

To explain the Sagnac effect, consider a light beam split into two beams that are constrained to follow the same closed path but in opposite directions where one can perform interferometer on the interacting beams when they return back to source. Then, gyroscopic motion can be determined when at rest, the light beams take the same amount of time to traverse the closed path in either direction while when rotating, one beam of light has a longer path to travel than the other and takes longer. This difference in time results in a phase difference in the interference pattern in the interferometer due to Sagnac effect where the relative phases or positions of the interference fringes that are shifted by tracking loop offset are proportional to the rotational rate of closed path, e.g. out of or into the page as shown above. Thus, three different loops would be required for a three-axis gyroscope.

Vibrating structure gyroscopes use the **Coriolis effect** to sense angular motion. Thus, these gyroscopes are also known as **Coriolis vibratory gyroscope (CVG)**. There are five general types of CVGs.

- Cylindrical Resonator Gyroscope (CRG)
- Piezoelectric gyroscopes
- Tuning fork gyroscope
- Wine-glass resonator
- Vibrating Wheel Gyroscope

For each of these types of CVGs, there are versions that can be made into microscopic devices incorporating both electronic and moving parts known as **micro-electromechanical systems (MEMS)**.

Inertial Measurement Units

Inertial measurement units (IMU) are multi-sensor electronic devices which measure the **specific force**, i.e. the non-gravitational force per mass and the angular velocity. For the three-dimensional vector quantities, an IMU uses three orthogonal accelerometers and three orthogonal gyroscopes. An IMU also typically has at least a simple onboard processor to use as a digital interface, long-term memory for unit conversion, and a temperature sensor for calibration modeling. Notably, there are many different IMU technologies for the underlying IMU design with vastly different performance grades.

Modern mechanical IMUs are often small **micro-electro-mechanical systems (MEMS)**. MEMS range in size from 0.004 to 1.0 mm² and are made up of components between 0.00001 to 0.01 mm², although components arranged in large arrays can be more than 1000 mm². They usually consist of an integrated circuit chip for simple data processing and microsensors, i.e. components that interact with the surrounding environment. Because of the large surface area to volume ratio of MEMS, forces produced by ambient electromagnetism, e.g. electrostatic charges and magnetic moments, and fluid dynamics, e.g. surface tension and viscosity, are more important design considerations than with larger scale mechanical devices. MEMS are manufactured by **lithographic construction**, i.e. the transfer of a pattern into a photosensitive material by selective exposure to a radiation source, e.g. light. The use of MEMS IMUs have allowed

IMUs to be made at the chip level and provide inertial information in a wide variety of applications beyond navigation.

IMU characteristics are often represented via acceleration and gyroscope error models based on the following sources which depend on various physical phenomena specific to each sensor technology:

- origin bias and axes misalignment: errors due to imperfect mechanical mounting
- incorrect scale factor: errors on first order sensitivity due to non-repeatabilities and nonlinearities
- sensor drift: errors between two measurements in similar conditions
- cross axis sensitivity: parasitic measurement error induced by solicitation along an axis orthogonal to sensor axis
- environment sensitivity: primarily thermal gradients and accelerations
- noise: dependent on desired dynamic performance

Depending on the targeted applications, the choice of IMU should consider the needs regarding stability, repeatability, and environment sensitivity, mainly thermal and mechanical environments, on both short and long time scales. However, as the raw sensor performance is repeatable over time, IMUs are typically compensated in real-time to enhance their performance based on both the sensor data and IMU models. The model complexity is chosen according to the needed performance and the type of application considered and are computed through a dedicated calibration sequence using multi-axis turntables and climatic chambers, e.g. a **tumble test**. They can either be computed for each individual product or generic for the whole product line. Calibration will typically improve a sensor's raw performance by at least two decades. Due to these production and calibration factors, the price and performance of IMUs vary widely. Thus, it is important to understand the different IMU performance grades in order to use the right sensor for an application.

There are roughly four grades of IMUs, namely

- **Navigation grade**, also known as **marine grade**:
 - ~ >\$50,000
 - ~ 0.5 m after 1 minute of integration
- **Tactical grade**:
 - ~ \$5,000-50,000
 - ~ 5 m after 1 minute of integration
- **Industrial grade**:
 - ~ \$100-1,000
 - ~ 50 m after 1 minute of integration
- **Consumer grade**, also known as **automotive grade**:
 - ~ \$10s
 - ~ 500 m after 1 minute of integration

18.2 Radionavigation Systems

Navigation satellite systems (NSS) are a specific type of radionavigation/radio-positioning system which transmit timed radio signals from artificial satellites orbiting Earth, known as a **satellite constellation**. In the future, one could also imagine using a satellite constellation around other planetary bodies which would assist in navigation on the surface of that body. As with most space technologies, these types of systems are notably expensive to launch as satellites are not cheap, especially those that must provide necessary information for navigation, i.e. positions of the broadcasting satellites, the timing of transmission, and the error models of significant system parameters. Thus, only governments have built NSS. There are three general types of NSS: global, regional, and augmentation.

There are currently four **global navigation satellite systems (GNSS)** constellations operating. The first GNSS installed was the United States' **Global Positioning System (GPS)** which is composed of 31-33 satellites stationed in 6 orbital planes. GPS became globally available in 1994. The second installed constellation of GNSS was Russia's **GLObal NAVigation Satellite System (GLONASS)** which is composed of 24-26 satellites stationed in 3 orbital planes. GLONASS was fully operational in 1995 over Russia, but reached a period of decline soon thereafter, but became globally available in 2016. The third constellation was the European Union's **galileo positioning system (GALILEO)** composed of 26 satellites in 3 orbital planes and became globally available in 2020. The fourth constellation was China's **BeiDou navigation satellite System (BDS)**, which is composed of 35-40+ satellites in 7 orbital planes and became globally available in 2020.

There are currently two **regional navigation satellite systems (RNSS)** constellations operating: India's **NAVigation with Indian Constellation (NAVIC)** with 7 satellites which became available in 2019 and Japan's **Quasi-Zenith Satellite System (QZSS)** composed of 4 satellites in elliptical GSO which became fully operational in 2018 with plans in 2023 to become independent of GPS with 7 satellites as an RNSS.

There are currently four **satellite-based augmentation systems (SBAS)** currently operating: the United States's **Wide Area Augmentation System (WAAS)**, the European Union's **European Geostationary Navigation Overlay Service (EGNOS)**, India's **GPS-aided GEO augmented navigation (GAGAN)**, and Japan's **Multi-functional Satellite Augmentation System (MSAS)**. SBAS only exist for enhancing the capabilities and performance of GPS by broadcasting additional corrections through GPS wavelengths to end users. These corrections are computed by ground station networks which use an uplink to the communication satellite constellation to broadcast corrections to users over a large region of the Earth's surface. **Ground-Based Augmentation Systems (GBAS)** which also exist for GPS, but use ground stations to broadcast computed corrections to GPS users within a smaller region than SBAS, but with potentially better corrections.

The history of NSS is relatively short in the history of navigation, but in recent times has become a ubiquitous technology in society. Below is an abbreviated timeline of its history.

- 1973: GPS proposed and funded by U.S. Department of Defense (DoD) as navigation technology for the military
- 1976: GLONASS proposed and funded by USSR
- 1978: GPS fully operational
- 1980: GPS civilian use allowed for signals with much poorer accuracy than military signals

- 1994: GPS globally available
- 1995: GLONASS fully operational at 24 satellites
- 1999: US military selectively denies access to India during Kargil War
- 2001: GLONASS at 6 operational satellites
- 2002: European Union (EU) and European Space Agency (ESA) agree to fund Galileo GNSS
- 2003: China joins the Galileo project
- 2003: Federal Aviation Administration (FAA) commissions WAAS
- 2006: China opts instead to upgrade then-regional BDS instead of joining Galileo GNSS
- 2006: WAAS operational
- 2008: GLONASS at 12 satellites
- 2009: ESA commissions EGNOS
- 2012: EGNOS operational
- 2016: GLONASS globally available at 24 satellites
- 2019: Galileo fully operational
- 2020: BDS globally available
- 2020: Galileo globally available

NSS are complex systems made up of three segments which will be described in the rest of this section. The first is the **space segment** which contains the artificial satellites in the constellations, also known as its **space vehicles (SV)**. The second is the **ground segment** which consists of the master control station (MCS) and monitor stations of the space vehicles. Lastly, there is the **user segment** which is made up of millions of users using a wide variety of GNSS navigation devices which at the most basic level must have the following components: an antenna, a receiver-processor, a highly stable clock, and relevant PNT algorithms.

Space Segment

In general, geocentric satellites can have a variety of orbits, but one way to characterize them is by orbital altitude, of which there are four types. **Low Earth orbits (LEO)** have altitudes below 2,000 km. **Medium Earth orbits (MEO)** have altitudes between 2,000 and 35,786 km. **Geosynchronous orbits (GSO)** are a unique orbit altitude at 35,786 km as since the satellites are orbit at the same rate as the average rotation of the Earth, i.e. one rotation per day. Furthermore, a specific GSO is the **geostationary orbit (GEO)** where the satellite remains stationary overhead of a point on the Earth's equator, thus these are highly useful for communication satellites to service particular locations on Earth. Finally, **high earth orbits (HEO)** have altitudes above 35,786 km.

For GNSS satellites, orbits near 20,200 km, i.e. MEO, are prescribed. This results in an orbital period of 12 hours and allows satellites to travel straight above the same location twice per day. Notably some GNSS orbits are also at GSO or GEO as well as the RNSS and SBAS orbits. The orbits and number of satellites in the constellation are designed to provide enough measurements for pseudorange multilateration.

The motion of satellite orbits is governed by the gravitational force of celestial bodies which is typically modeled by **Newton's gravitational law**

$$F_g = G \frac{m_{\text{Earth}} m_{\text{SV}}}{\|\vec{x}^E\|_2^2} \quad (18.6)$$

where G is the gravitational constant. Notably Newtonian gravity neglects General Relativity which has a small, but noticeable effect on space vehicles. This model for the acceleration induced by the celestial body, the motion then depends on the initial velocity which dictates whether the orbit will be an ellipse, parabola, or a hyperbola, of which the last two model the eventual escape of the space vehicle from the influence of the celestial body. For elliptical orbits, the solution for a satellite's motion due to one body can be described using Kepler's laws of planetary motion which originally applied to natural satellites of the Sun.

Control Segment

The control segment of a GNSS is made up of a network of monitor stations with a centralized **master control system (MCS)** which for GPS is located at Schriever Space Force Base (SFB) in Colorado. The MCS has many duties to perform as part of the overseer of a GNSS including commanding the space vehicles to perform any orbital manuevers, estimating the navigational state of the satellites, generating the timing and navigational messages in the GNSS signal(s), monitoring the system integrity, and updating the processor programs of the GNSS satellite digital systems if necessary. In order to obtain information about the status of the satellites for the purposes of command and control, the MCS collects data from **monitor stations** around the globe. GPS has six monitor stations. These stations track and collect the GNSS signals and local atmospheric data and feed this information directly to the MCS for analysis. In addition, to communicate with the GNSS satellites, the MCS uses ground antennas stationed around the globe to uplink with the satellites, i.e. send commands. GPS has four ground antennas.

As an operational system, there are several factors that must be continuously monitored. The first is that any of the various subsystems of the satellites will have different effects on the observable behavior of the satellites. The control segment monitors the satellites for a variety of faults or errors which make using GNSS unsafe for users. Secondly, the satellite clocks are not perfect and therefore the control segment must monitor and update the clocks to provide accurate timing for radionavigation. Another major consideration is the perturbations in the satellites' orbits caused by gravitational effects and space weather. The MCs estimates the navigational state of each satellite and can command corrective maneuvers if required to keep the constellation within operational limits. It also can command orbital maneuvers to de-service satellites when they are no longer operational.

User Segment

The original users of GPS and GNSS were the military who desired a global solution to navigation for command and control of their assets. This is still a major part of the GNSS systems which contain encrypted

military channels that enable high precision GNSS navigation. However, civilian navigation is still possible for pedestrians and a variety of vehicles including automobiles, aircraft, cyclists, ships, and even spacecraft in LEO. Some other uses for GNSS include mapping the Earth (e.g. Geographic Information Systems (GIS), archaeology, and surveying), synchronized timing (e.g. cell networks), mobile satellite communications (provides antenna pointing information in the global ECEF frame), weather prediction (infers atmospheric conditions from GNSS radio signals), and much more.

However, GNSS still has some shortcomings and challenges for navigational use in all circumstances, the primary being a requirement for a clear Line-of-Sight (LOS) to the satellites for the signal to be measurable by GNSS antennas. This causes users to lose the signal indoors and in canopies or canyons, either natural or urban, or which the latter is a large area of need for reliable navigation in the near future. Another challenge in GNSS is that atmospheric transmission introduces significant error which has been a major subject of research in 2000-2010s. Third, is the possibility of jamming the GNSS signal due to its low power being susceptible to any elevated noise which can be intentional or unintentional. Lastly, there is the threat of **spoofing**, i.e. creating a false GPS signal which replicates the expected external radio signal. This can make receiver estimate completely a wrong trajectory. This last subject matter is of critical importance in the modern age of connected systems and thus there is a lot of current research into **Resilient PNT** to develop algorithms and systems to overcome system attacks.

The International GNSS Service (IGS) publishes standards for communicating GNSS information. The primary format is the **receiver independent exchange (RINEX)** which has a standard for storing the observation data, the navigation message, and the meteorological data. The RINEX format was adopted by the GNSS community to catalog raw GPS data for post-processing by any potential user. This allows more potentially more accurate positioning to be done with information better than or unknown to the original receiver, e.g. better atmospheric or orbital models than at the actual time of measurement. This format can be processed by a multitude of applications by using standard information parameters. This format has evolved over time with the development of GPS technologies with the most recent being RINEX 3. It should also be noted that often RINEX files are compressed as ASCII-based CompactRINEX or CRINEX using the Hatanaka compression scheme. As such, there are open-source parsers available, e.g. for python the `georinex 1.14.1` package is available.

The observation data contains the observations or measurements from all GNSS frequencies and all satellites. This includes the pseudoranges, Doppler shift and the carrier phase which will be discussed later in the course. The navigation message contains the ephemeris information from the MCS broadcasted model while the IGS precise orbit solutions are published in the *sp3* format. The IGS also publishes other supplementary materials for GNSS data processing. This includes IGS monitoring station position and velocity solutions which are published in the Station Independent Exchange (SINEX) format, the Earth rotation parameter (erp) files, the ionospheric exchange (IONEX) format for TEC grid products, and the Zenith path delay products computed at each IGS monitoring station in the format Tropo SINEX.

GNSS Time Standards

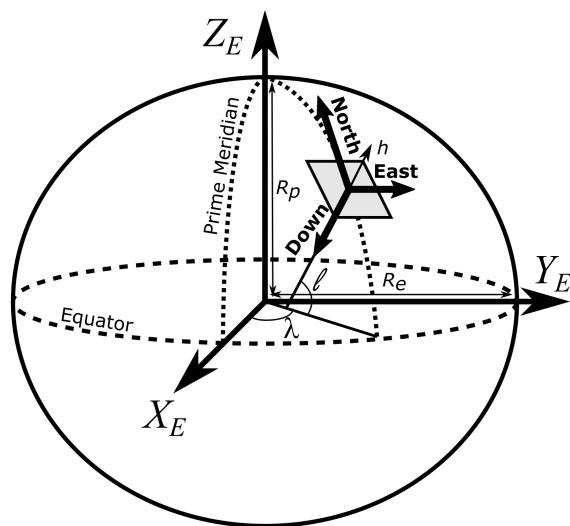
A second important piece of GNSS are the individual time standards used by each GNSS constellation. GPS uses **GPS Time (GPST)** as a continuous time scale and theoretically accurate to about 14 ns. However, most receivers lose accuracy in the interpretation of the signals and are only accurate to 100 ns. The GPST differs from TAI and UTC, but remains a constant 19 seconds behind TAI as it does not implement leap seconds.

Periodic corrections are performed to the on-board clocks in the satellites to keep them synchronized with ground clocks. The GPS navigation message includes the difference between GPST and UTC. GPST 0 occurred at 12:00AM on January 6, 1980. As of July 2015, GPST is 17 seconds ahead of UTC because of the leap second added to UTC on June 30, 2015. GLONASS uses **GLONASS Time (GLONASST)** as generated by the GLONASS Central Synchroniser and is typically accurate to within 1,000 ns. Unlike GPS, the GLONASS time scale implements leap seconds, like UTC. Galileo uses **Galileo System Time (GST)** as a continuous time scale which is generated on the ground at the Galileo Control Centre in Fucino, Italy, by the Precise Timing Facility, based on averages of different atomic clocks and maintained by the Galileo Central Segment and synchronized with TAI with a nominal offset below 50 ns. According to the European GNSS Agency Galileo is accurate to about 30 ns. The Galileo navigation message includes the differences between GST, UTC and GPST. BeiDou uses **BeiDou Time (BDT)** as a continuous time scale starting at 1 January 2006 at 0:00:00 UTC and is synchronized with UTC within 100 ns.

GPS satellites have at least two onboard caesium and as many as two rubidium atomic clocks. Each Galileo satellite has two passive hydrogen maser and two rubidium atomic clocks for onboard timing. The vast majority of GNSS receivers use quartz clocks, typically symbolized by *OCXO*. Quartz clocks are used because they are relatively inexpensive, compact, use low amounts of power, and possess a long life, but are not at the atomic clock standard used by the GNSS SV. However, more expensive receivers may allow an external atomic timing source, e.g. a caesium or rubidium clock.

WGS Earth Reference Frame

A third important piece of GNSS is the use of reference frames, namely, ellipsoidal geodetic coordinates, LLA, and NED coordinates.



GNSS systems use the **WGS84 parameters** which defines the following constants for Earth. The ellipsoidal **equatorial radius** is given in meters as

$$R_0 = 6378137 \quad (18.7)$$

The ellipsoidal **flattening** is given as

$$f = \frac{1}{298.257223563} \quad (18.8)$$

The **Earth angular rate** is given in rad/s as

$$\omega = 7292115 \times 10^{-11} \quad (18.9)$$

The **Earth gravitational parameter** for GPS is given in m^2/s^2 as

$$\mu_E = 3.9860050 \times 10^{14} \quad (18.10)$$

Notably, one can derive the ellipsoidal **polar radius** in meters as

$$R_P = R_0(1 - f) \quad (18.11)$$

and the ellipsoidal **eccentricity**

$$e = \sqrt{f(2 - f)} \quad (18.12)$$

GPS Signal Design

GPS satellites, a.k.a. **space vehicles (SV)** in GPS documentation broadcast the GPS signals as **right hand circular polarized (RHCP)** signals, i.e. the electromagnetic wave is polarized in a direction that oscillates in a continuous rotating fashion with a rotation defined by the right hand rule. In particular, the electrical field vectors of GPS signals have constant magnitude, but the electric field traces out a helix in direction of propagation.

The GPS carrier signals are broadcast on multiple frequencies. There are five GPS frequencies all within the **L-band** of the radio spectrum, i.e. 1 to 2 gigahertz (GHz), three of which are used as navigation frequencies. The L1 GPS carrier is broadcast at 1575.42 MHz and carries the L1C or *Civilian*, also known as the *Course/Acquisition (C/A)*, the encrypted *Precision* or P(Y) code used by the U.S. DoD, and a newer encrypted DoD *Military-code (M-code)*. The L2 GPS carrier is broadcast at 1227.60 MHz, and carries the L2C or *Civilian*, the second piece of the *Precision* P(Y), and the second piece of the *Military-code (M-code)*. The L5 GPS carrier is broadcast at 1176.45 MHz and is used for safety-of-life (SOL) information. It should be noted that the L3 & L4 carriers do not provide information for navigation, but instead are used for nuclear warhead enforcement and atmospheric research, respectfully.

GPS satellites in GPS documentation transmit modulated radio signals use multi-frequency carrier signals encoded with digital code through **binary code modulation (BCM)** at a much slower frequency than the carrier. These signals are broadcast in a cone of approximately 28° , which at a MEO of 20,200 km, this is enough to cover one-side of the Earth's geoid. BCM encodes 0's and 1's on the carrier signal through a variety of techniques including **amplitude modulation (AM)** for analog signals or **amplitude-shift keying (ASK)** for digital signals, **frequency modulation (FM)** for analog signals or **frequency-shift keying (FSK)** for digital signals, and **phase modulation (PM)** for analog signals or **phase-shift keying (PSK)** for digital signals. Of these, PSK has become a very popular form of radio communication technology including wireless local area networks (WLAN), radio frequency identification (RFID), Bluetooth, and GNSS. In addition, another form of modulation developed in particular for GNSS is **binary offset carrier (BOC) modulation**, also known as **split spectrum modulation**.

GPS phase modulation uses several techniques in order to provide a *spread spectrum* technique on the carrier in order to achieve better signal-to-noise ratio (SNR), provide more accurate ranging, less radio frequency interference (RFI), and increased security. However, this technique also weakens the signal tremendously, so much so that sensing the GPS signal is often compared metaphorically to identifying a 25-watt light bulb from 10,000 miles away.

As GPS signals are broadcast from multiple satellites at the same frequencies, GPS uses **code division multiple access (CDMA)** to track each signal. CDMA allows multiple operating SVs with unique IDs (SV ID) to be tracked where each SV ID broadcast is a unique pseudorandom noise (PRN) number, a predictable signal, but statistically resembles noise where each PRN does not correlate well with each other allowing for correlation tracking methods to be used to distinguish SVs. Lastly, the GPS signals each carry a navigation message modulated on the PRN and carrier which is much slower than both. For GPS, the civilian codes are modulated using PSK (PSK) for the L1 C/A and L2C which modulates the code between 2 phases separated by 180° while BOC is used for the L1C signal to allow its simultaneous transmission at L1 with low interference with BPSK which has a centered peak around the carrier frequency.

As GPS depends on accurate timing of the signal transmissions and uses precise signal processing to overcome its low signal power, all components of GPS digital systems are multiples of a standard rate known as the **fundamental clock rate**, denoted by F_0 , and equal to 10.23 MHz. This is set by caesium and rubidium clocks onboard the SVs. Of note, the three navigation L-band carriers are set at $154F_0$ for L1, $120F_0$ for L2, and $115F_0$ for L5. However, the BCM signals are much slower than the clock rate, e.g. the L1 C/A PRN oscillates at 0.1, $F_0 = 1.023$ Mbps, and the L1 C/A navigation message oscillates at 50 bps. SVs experience relativistic effects which cause SV clocks to appear to run $38 \mu\text{s}/\text{day}$ faster rate than Earth clocks. As SVs travel at roughly 3.874 km/s, this produces a special relativistic effect on the SV clocks as appearing to run slower than Earth clocks at roughly $-7 \mu\text{s}/\text{day}$. In addition, the weaker gravity generates a general relativistic effect which causes the SV clocks to appear to run faster than earth clocks at about $45 \mu\text{s}/\text{day}$. Together, these create an overall . Thus, to produce $F_0 = 10.23$ MHz in space, the SV clocks are set to 10.22999999543 MHz.

GPS Signal Processing

The design of the GNSS signal greatly reduces the power of signal which causes the GNSS signals to be less than the natural receiver noise. This is notably the opposite of directional communication satellites which use large disk antennas to focus the signal on both transmit and receive. However, GPS signals are made observable through specialized signal processing. GNSS receiver-processors sense the radio signal using an antenna and process the complex radio signal to extract the relevant PNT information. The antenna serves to convert the radio waves to an electrical current and must have high sensitivity, i.e. gain, for which different types of antennas have been used for different applications and include patch, dipole, quadrifilar, and helix. Due to the wide broadcast beamwidth, these antennas are designed to be omni-directional. Once the signal has been converted to a current, the GPS receiver-processor can process the signal by either radio circuitry, i.e. a **hardware-defined radio (HDR)**, or digital signal processing, i.e. a **software-defined radio (SDR)**. GPS receiver-processors can have a wide variety of sizes and components, but require at least a pre-amplifier, a bandpass filter, a phase-lock loop (PLL), carrier signal demodulation, and a delay-lock loop (DLL).

As the SVs orbit the Earth, the observed carrier frequencies will increase and decrease by up to 5-10 kHz due to the Doppler effect. This change, known as the **Doppler shift** can be predicted as the SV pass across the

sky. However, not only do the SVs move, but the GPS receiver may also move which will cause an additional Doppler shift which the receiver must detect and track. This Doppler shift captures the relative line-of-sight (LOS) motion between each SV and the receiver and can be used as the **GPS range rate measurement** in addition to the signal processing of the GPS signal. In order to acquire and track the Doppler shift, the receiver-processor replicates the original frequency carrier and differences it with the observed signal from the antenna. This new differenced signal oscillates with an **intermediate frequency (IF)**, also known as the **beat frequency**. The beat frequency error is then detected and tracked by a PLL in the receiver-processor which allows the carrier signal to be tracked accurately and forms the **GPS carrier phase measurement**. Finally, the receiver-processor demodulates the carrier signal from the encoded signal before passing it to the next tracking stage.

Next, the receiver-processor needs to acquire and track the encoded signals *from each SV*. To do so, the receiver-processor generates the unique PRNs for each SV ID. Recall that the each PRN does not correlate well with other PRNs, i.e. *orthogonal* signals, thereby allowing the receiver-processor to distinguish the different SV signals. For example, the L1C PRN codes consist of 1023 chips and are generated by an *exclusive or* combination of two 10-stage linear feedback shift registers (LFSR), i.e. bit-streams determined by linear function of certain stages, i.e. “taps”. The uniqueness of each L1C PRN is performed by a unique integer delay on second LFSR. Similar to the Doppler shift, the receiver-processor can acquire each PRN by generating replica PRNs and by shifting the PRNs bit-by-bit until one finds the “best fit” from all satellite PRNs, a process also known as **PRN auto-correlation** where matching the signals in time will theoretically create a signal correlation of 1 though this will be slightly lower in practice due to processing errors. Receiver-processors typically use a Fast Fourier Transform (FFT) to perform an efficient and accurate PRN auto-correlation.

After acquiring the PRN signal, a DLL tracks the PRN code shift over time to provide the **GPS pseudorange measurement** as the measured time shift that provides the maximum correlation. This is also known as the **code phase measurement**. In addition, this tracking allows the receiver-processor to read the much slower navigation message that still remains on the signal. The DLL is typically a digital proportional-integral (PI) controller that compares the phase of its last output with the input clock which it integrates and feeds back as the control to all delay elements. Furthermore, to acquire and track multiple satellites simultaneously, receiver-processors run multiple tracking channels in parallel. In particular, each single frequency from a single satellite has its own dedicated channel which notably requires hundreds of channels in multi-frequency, multi-constellation GNSS receiver-processors. This paradigm allows the receivers to maintain accuracy when moving, anti-jamming capability, and shortens the time to first fix on the PRNs. Thus, each channel has two modes of operation: *acquisition* which performs the PRN auto-correlation calculation and *tracking* which uses the DLL to track the slightly varying time shift to obtain the code phase measurement.

GPS Navigation Message

There are three specific formats for the GPS navigation message encoded on the civilian signals. However, each format of the GPS navigation message contains at least the following:

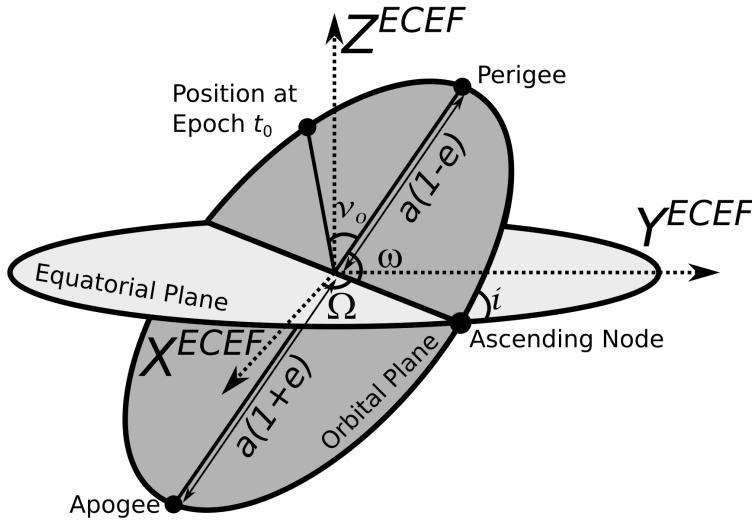
- **Almanac**
 - Information for every SV

- Atmosphere model
 - Relate GPS time to UTC
 - Updated every 6 days
- **Broadcast Ephemeris**
 - Binary message in SV signal
 - Updated every 2 hours by MCS

The L1 C/A Legacy Navigation (LNAV) message which supports 32 SVs and has five sub-frames. The first frame contains information on the satellite clock and GPS Time. The second and third contains broadcast ephemeris for the particular satellite, and the fourth and fifth contain the almanac which provides coarse orbital elements for *all* SVs, a global ionospheric model, and GPS time relationship to UTC. The entire almanac takes 25 frames to complete (e.g. 12.5 minutes), but is used by receivers to acquire a set of viewable SVs which can be used for replicate PRNs for receivers with a low number of channels. However, once acquired, the specific SV broadcast ephemeris is used for positioning. The L1/2C: Civilian Navigation (CNAV/CNAV-2) which supports 63 SVs and has 12-second 300-bit messages which are analogous to LNAV sub-frames, but with additional information. In particular, this format contains a GPS-to-GNSS time offset which facilitates better interoperability with other GNSS, extra bandwidth which enables a differential correction which can correct the L1 navigation clock, and an alert flag which can be set by the MCS if the SV data cannot be trusted. This alert flag is a much more rapid notification than the LNAV message which is necessary for safety-of-life applications using GPS. Lastly, the GPS time format is also limited within each navigation message format. The LNAV expresses GPST as the *week* and *time of week (TOW)* which has a resolution of 1.5 seconds. The CNAV/CNAV-2 expresses GPST as the *interval time of week (ITOW)* and *time of interval (TOI)* to increase the bit length of GPS time. For LNAV, GPST rolls back to 0 every 1,024 GPS weeks (19.6 years) while CNAV/CNAV-2 GPST rolls back to 0 every 8192 weeks (157.0 years).

GPS Satellite Positions from Ephemeris

The most crucial factor in GNSS pseudorange multilateration is determining the position of the transmitting artificial satellites. Thus, this subsection will cover an algorithm for converting the **GPS ephemeris**, i.e. a quantitative description of the positions, and possibly velocity, of satellites over time, to SV ECEF positions based on their elliptical orbits



where the labeled GPS ephemeris parameters are the six **Keplerian elements** of the orbit, i.e.

- the **eccentricity**, $0 < e < 1$, which is a measure of the elongation of the elliptical orbit compared to a circular orbit ($e = 0$),
- the **semimajor axis**, a , which is the distance between the perigee and apogee, i.e. the closest and further points from the focus corresponding to the Earth's center of gravity,
- the **inclination angle**, i , which is the angle between the equatorial and orbital planes at the ascending node where the space vehicle moves from beneath the equatorial plane to above,
- the **longitude of ascending node**, Ω , which is the angle relative to the prime meridian of Earth where the SV intersects the equatorial plane,
- the **argument of perigee**, ω , which is the angle in the orbital plane from the ascending node to perigee,
- the **true anomaly at epoch**, ν_0

To compute an SV's ECEF position using the Keplerian elements, one must derive the true anomaly, ν , from the eccentric anomaly, E , and the current mean anomaly, M . To do so, one must use M to solve for the eccentric anomaly E using **Kepler's equation**

$$M = E - e \sin E \quad (18.13)$$

which has no analytical solution and must be solved numerically. Next, one can compute ν using

$$\nu = \text{atan2} \left(\frac{\sqrt{1 - e^2} \sin E}{1 - e \cos E}, \frac{\cos E - e}{1 - e \cos E} \right) \quad (18.14)$$

Then, one can convert the orbital angles to the position along the elliptical orbit and, finally, compute the ECEF position by a frame transformation using (i, Ω, ω) as Euler angles for elliptical orbit.

From the broadcasted navigation message, one can parse out the following information:

- t_0 : Reference time of ephemeris
- Keplerian Elements
 - \sqrt{a} : square root of semimajor axis
 - e : eccentricity
 - i_0 : inclination angle (at time t_0)
 - Ω_0 : longitude of ascending node (at weekly epoch)
 - ω : argument of perigee (at time t_0)
 - M_0 : mean anomaly (at time t_0)
- Rates of change of select Keplerian elements
 - IDOT: di/dt
 - OMEGADOT: $\dot{\Omega}$
 - Δn : mean motion correction
- Amplitude of sinusoidal corrections
 - C_{uc}/C_{us} : cosine/sine correction to argument of latitude
 - C_{rc}/C_{rs} : cosine/sine correction to orbital radius
 - C_{ic}/C_{is} : cosine/sine correction to inclination angle

As part of the algorithm, one must also use the following GPS-defined constants

- The WGS84 value of Earth's gravitational parameter

$$\mu = 3.986005 \times 10^{14} \text{ m}^3/\text{s}^2 \quad (18.15)$$

- GPS value for speed of light

$$c = 2.99792458 \times 10^8 \text{ m/s} \quad (18.16)$$

- WGS84 value of Earth's rotation rate

$$\dot{\Omega}_e = 7.2921151467 \times 10^{-5} \text{ rad/s} \quad (18.17)$$

- GPS value for π

$$\pi = 3.1415926535898 \quad (18.18)$$

Alternatively, one can also compute the precise satellite positions from the orbital products of the IGS which use the IGS08 reference frame. This frame was first defined on January 1, 2005 and, similar to the WGS84, the IGS08 uses the Geodetic Reference System (GRS) 1980 ellipsoid with the Greenwich prime meridian and an origin at geocenter. It is derived from a subset of the 232 stable IGS station coordinates at epoch 2005.0 and is practically coincident with ITRF2008 and WGS84, all of which agree at the centimeter level which is also the uncertainty magnitude for each frame solution.

18.3 Radar, Electro-Optical, and Infrared Sensors

Energy wave sensors use electromagnetic or acoustic signals to form measurements and these can be **active**, i.e. the energy is generated by the sensor, or **passive**, i.e. the energy is generated by the environment. Active energy wave sensing uses active sonar, radar, and lidar while passive energy wave sensing uses passive sonar and optical cameras. These energy wave sensors can produce one-, two-, or three-dimensional signals depending on the type and number of sensors, where two- or three-dimensional signals are also known as **images**.

Radar Sensors

Radio detection and ranging (radar) is a remote sensing technique that uses active radio wave transmission and reception to derive pseudorange, pseudoazimuth, pseudoelevation, and pseudorange rate measurements. Radar use antennas to both transmit and receive the radio signals. When a single antenna is used for both, one has a **mono-static radar**. When separate transmitter and receiver antennas are used, one has a **bi-static radar**. When three or more transmitters and receivers are used with spatial separation, one has a **multi-static radar**.

When radio waves contact a target, also known as **illumination**, they are usually reflected or scattered in many directions, although some of them will be absorbed and penetrate into the target. Radar signals are reflected especially well by materials of considerable electrical conductivity—such as most metals, seawater, and wet ground. If electromagnetic waves traveling through one material meet another material, having a different dielectric constant from the first, the waves will reflect or scatter from the boundary between the materials. This means that a solid object in air or in a vacuum, or a significant change in atomic density between the object and what is surrounding it, will usually scatter radio waves from its surface. This is particularly true for electrically conductive materials such as metal and carbon fiber, making radar well-suited to the tracking of flight vehicles, ground vehicles, and surface vessels.

Radar waves scatter in a variety of ways depending on the wavelength of the radio wave and the shape of the target. If the wavelength is much shorter than the target's size, the wave will bounce off in a way similar to the way light is reflected by a mirror. If the wavelength is much longer than the size of the target, the target may not be visible because of poor reflection. The most reflective targets for short wavelengths have 90° angles between the reflective surfaces. A **corner reflector** consists of three flat surfaces meeting like the inside corner of a box. The structure will reflect waves entering its opening directly back to the source. They are commonly used as radar reflectors to make otherwise difficult-to-detect targets easier to detect. Conversely, targets intended to avoid detection will not have inside corners or surfaces and edges perpendicular to likely detection directions, e.g. “odd” looking stealth aircraft. The extent to which a target reflects or scatters radio waves is called its **radar cross section**, σ .

The **radar range equation** provides an equation for the power returning to the receiving antenna, P_r , as a function of the transmitter power, P_t , as

$$P_r = \frac{G_t A_r \sigma F^4}{(4\pi)^2 R_t^2 R_r^2} P_t \quad (18.19)$$

where G_t is the gain of the transmit antenna, A_r is the effective aperture of the receiving antenna, F is the pattern propagation factor which accounts for multipath and shadowing, R_t is the distance from the transmitter to the target, and R_r is the distance from the target to the receiver. Notably, one can also

$$A_r = \frac{G_r \lambda^2}{4\pi} \quad (18.20)$$

where λ is the transmitted wavelength and G_r is the gain of the receiving antenna. Also, if the transmitter and receiver are the same antenna, then $R_t = R_r = R$ and

$$P_r = \frac{G_t A_r \sigma F^4}{(4\pi)^2 R^4} P_t \quad (18.21)$$

which shows the distant target received power is relatively very small.

The **radar Doppler shift**, f_D , is related to the transmit frequency, f_t as

$$f_D = 2 \frac{\dot{\rho}}{c} f_t \quad (18.22)$$

where $\dot{\rho}$ is the range rate between the target and radar. Of note, when the transmit frequency is pulsed at a frequency of f_r , the resulting frequency spectrum will contain harmonic frequencies above and below f_t with a distance of f_r . Thus, the Doppler shift is unambiguous if

$$|f_D| < \frac{1}{2} f_r \quad (18.23)$$

where $f_r/2$ is known as the **Nyquist frequency** since the returned frequency cannot be distinguished from the shifting of a harmonic frequency above or below. When substituting with f_D , one has

$$|\dot{\rho}| < \frac{c f_r}{4 f_t} \quad (18.24)$$

Shot noise is produced by electrons in transit across a discontinuity, which occurs in all detectors. Shot noise is the dominant source in most receivers. There will also be **flicker noise** caused by electron transit through amplification devices, which is reduced using **beat frequency** amplification of mixed signals. Another reason for beat frequency processing is that for fixed fractional bandwidth, the instantaneous bandwidth increases linearly in frequency which improves range resolution. The one notable exception to beat frequency radar systems is **ultra-wideband (UWB) radar**. Noise is also generated by external sources, most importantly the natural thermal radiation of the background surrounding the target of interest. In modern radar systems, the internal noise is typically about equal to or lower than the external noise.

The **thermal noise** is given by $k_B T B$, where T is temperature, B is bandwidth (post-matched filter) and k_B is Boltzmann's constant. Matched filtering, i.e. allows the entire energy received from a target to

be compressed into a single range, Doppler, elevation, or azimuth bin. While it appears this would fixed interval of time one could obtain perfect, error-free, detection. To do this one simply compresses all energy into an infinitesimal time slice. However, while time is arbitrarily divisible, current is not as the quantum of electrical energy is an electron, and so the best one can do is match filter all energy into a single electron. Since the electron is moving at a certain temperature, i.e. the Planck spectrum, this noise source cannot be further eroded.

One way to obtain a range measurement is based on the time-of-flight (TOF), i.e. transmit a short pulse of radio signal and measure the time it takes for the reflection to be received. The range to the target is one-half the round trip time multiplied by the speed of the signal. Through the use of a **duplexer**, a mono-static radar switches between transmitting and receiving at a predetermined rate. A similar effect imposes a maximum range as well. In order to maximize range, longer times between pulses should be used typically specified as smaller **pulse repetition frequencies (PRF)**. These two effects infer good short range and good long range measurements are not possible in a single radar. This is because the short pulses needed for a good minimum range broadcast have less total energy, making the returns much smaller and the target harder to detect. This could be offset by using more pulses, but this would shorten the maximum range. So each radar uses a particular type of signal. Long-range radars tend to use long pulses with long delays between them, and short range radars use smaller pulses with less time between them. As electronics have improved many radars now can change their PRF, thereby changing their range. Modern advanced radars fire two pulses during one cell, one for short range and one for long range.

Another form of distance measuring radar is based on **frequency modulation (FM)**. In these systems, the frequency of the transmitted signal is changed over time. Since the signal takes a finite time to travel to and from the target, the received signal is a different frequency than what the transmitter is broadcasting at the time the reflected signal arrives back at the radar. By comparing the frequency of the two signals the difference can be easily measured. A further advantage is that the radar can operate effectively at relatively low frequencies. This technique can be used in **frequency modulated, continuous wave (FMCW) radar** where a carrier signal is frequency modulated in a predictable way, typically varying up and down with a sine wave or sawtooth pattern. This signal is sent out from one antenna and received on another and continuously compared using a simple beat frequency modulator that produces a frequency tone from the returned signal and a portion of the transmitted signal. The modulation index riding on the receive signal is proportional to the time delay between the radar and the reflector. The frequency shift becomes greater with greater time delay. The frequency shift is directly proportional to the distance traveled.

The two techniques outlined above both have their disadvantages. The pulse timing technique has an inherent tradeoff in that the accuracy of the distance measurement is inversely related to the length of the pulse, while the energy, and thus direction range, is directly related. Increasing power for longer range while maintaining accuracy demands extremely high peak power. The FMCW methods spread this energy out in time and thus require much lower peak power compared to pulse techniques, but requires some method of allowing the sent and received signals to operate at the same time, often demanding two separate antennas. Thus, modern radar combine these two techniques using **pulse compression** which start with a longer pulse that is also frequency modulated. Spreading the broadcast energy out in time means lower peak energies can be used, but upon reception, the signal is sent into a system that delays different frequencies by different times. The resulting output is a much shorter pulse that is suitable for accurate distance measurement, while also compressing the received energy into a much higher energy peak and thus reducing the **signal-to-noise ratio (SNR)**.

To measure the bearing angles to a target, several receivers measure the relative arrival time to each which provides a bearing angle or with an array, one can measure the relative amplitude in the beams formed through a process called **beam-forming**. Advanced radars often have multiple beams to provide all-round cover while simple ones only cover a narrow arc, although the beam may be rotated by mechanical scanning. The target signal together with noise is then passed through various forms of signal processing, which for simple sonars may be just energy measurement, and one uses classic detection algorithms, e.g. hypothesis testing, to decides whether the output is the required signal or noise. Further processing may be done to classify the target.

Aperture radar use radar arrays to illuminate an entire region, known as an **aperture** whose output is a two- or three-dimensional image. Another technique is to use a single radar in motion as a **synthetic aperture radar (SAR)**. To create a SAR image, successive pulses of radio waves are transmitted to illuminate a target scene, and the echo of each pulse is received and recorded. The pulses are transmitted and the echoes received using a single beam-forming antenna with varying wavelengths. As the SAR moves, the antenna location relative to the target changes with time. Signal processing of the successive recorded radar echoes allows the combining of the recordings from these multiple antenna positions. This process forms the synthetic aperture and allows the creation of higher-resolution images than would otherwise be possible with a given physical antenna.

Electro-Optical/Infrared Sensors

Cameras, i.e. passive optical sensors, **light detection and ranging (lidar)**, i.e. active optical systems are remote sensing techniques that use the electromagnetic spectrum from infrared to visible to ultraviolet light to form images. To form a two-dimensional image, a **mono-vision camera** is used and typically is specified by the wavelength band or bands used, i.e. an **infrared camera**, **near-infrared camera**, **visual camera**, **multispectral camera** which uses 3-10 wide bands of infrared light, and **hyperspectral camera** which uses 100s of narrow bands of optical light. To form a three-dimensional image using vision, one may use a **stereo-vision camera**. If the three-dimensional image does any wavelength information about the received signal, then one has forms a **point cloud** in three-dimensions which can also be generated by a lidar.

For image processing and target detection of two- and three-dimensional images into range and bearing, **convolutional neural networks (CNN)** have become the *de facto* standard for **computer vision**, also known as **machine vision**, a field which encompasses the techniques and algorithms for processing images into other information including useful measurements for sensor fusion. These data-driven algorithms use truth databases to form object detection and measurements given the image information. From two-dimensional images, one can form bearing measurements as part of the CNN output. For three-dimensional images and point clouds, one can also obtain range measurements.

Radar, Lidar, and Optical Altimeters

Radar altimeters and **Lidar altimeters** measure the time of flight for a radio or light signal to reflect back from the ground and were developed as a more reliable and accurate sensor than barometric altimeters when heavy fog or rain is present. **Frequency Modulated Continuous-Wave (FMCW) radar** are more commonly used than pulsed radar since the frequency shift can be more accurately tracked through signal processing than the direct time difference. For FMCW radar, the greater the frequency shift, the further the distance traveled. In modern aircraft, radar altimeters are standard sensors for automated landing systems

and terrain-avoidance systems. In satellite systems, radar altimeters are also used in remote sensing of the Earth's surface.

These systems provide a relative **height above ground level (HAGL)** or **height above average terrain (HAAT)**. Thus, for an absolute altitude estimate, these sensors also require a **terrain map**, for the ground level or the average terrain. For the HAGL measurement, an altimeter uses a clock and the principle of two-way **time of flight (TOF)** measurements of the difference between the time of transmission (TOT) and the time of arrival (TOA) for the energy signal, i.e.

$$h = \frac{c(ToF)}{2} = \frac{c(TOA - TOT)}{2} \quad (18.25)$$

where h is the relative height and c is the speed of transmission.

Radar, Lidar, and Optical Odometers

18.4 Air Data Systems

Air data systems (ADS) are multi-sensor information systems that provide measurements of a flight vehicle's surrounding air mass, collectively known as the **air data**, typically quantified as the airspeed, altitude, rate of climb, and angles of attack and sideslip. ADS are safety-critical for the majority of airplane operations as air data measurements are often used in the gain scheduling of the guidance and control laws. These errors are typically failures of the ADS itself, classified as **system faults**, while other errors are the result of environmental factors, classified as **inherent errors**. **Synthetic air data system (SADS)** provide airspeed, angle of attack, and angle of sideslip, known as the **air data triplet**, typically using the wind triangle, and can be model-based or model-free. SADS are primarily used for risk monitoring of a real air data system.

For the altitude and airspeed, most ADS use the principle of an air pressure gradient, i.e. pressure differences, as measured from a pitot tube and a static port/source, separately, or together as a pitot-static system.



The pitot tube measures the **pitot pressure**, P_p , also known as the **ram pressure**, defined as the air pressure created by the aircraft motion or the air "ramming" into the tube. Under ideal conditions, the pitot pressure is equal to the **total pressure**, P_t , also known as the **stagnation pressure**. The pitot tube is most often mounted on front of the wing or the nose, facing forward, where its opening is exposed to the relative wind and is less impacted by the aircraft's structure. The static port measures the **static pressure**, P_s . Under ideal conditions, the static pressure is equal to the **atmospheric pressure**, P_a , which defines the nominal pressure of the surrounding air mass with no objects moving inside. The static port is often a flush-mounted hole on the fuselage of an aircraft where it can access the air flow in a relatively undisturbed area. This static port can also be used directly in a barometric altimeter.

The difference between the total and atmospheric pressures is called the **dynamic pressure**, Q , i.e.

$$Q = P_t - P_a = \frac{1}{2}\rho v_\infty^2 \approx P_p - P_s \quad (18.26)$$

which allows a direct computation of the airspeed *if* the air density is known. The air density typically obtained by ISA model with temperature, altitude, and static pressure. Some pitot-static systems may also directly measure the rate of climb by intentional releasing air from a static source port to a trailing air port. By measuring the rate of discharge through this leak relative to a calibrated nominal value, the rate of climb can be inferred.

To measure the angles of attack and sideslip, an ADS use differential pressure ports, wind vanes, and/or null-seeking pressure tubes. In many ADS the differential pressure ports are typically placed on a single pitot tube and commonly use a five-hole or seven-hole geometry. With this tube, one simple method about straight-and-level flight is to use the center port for the airspeed, top-bottom port difference for angle of attack, and the left-right port difference for the angle of sideslip. However, more accurate ADS use calibration tables or a polynomial fit to calibration tests using all ports together to obtain the air data triplet.

It should be noted that often the **flank angle** is measured by the airplane's air data system (ADS) instead of the sideslip angle, i.e.

$$\beta_f = \tan^{-1} \frac{v}{u} \quad (18.27)$$

which is approximately equal to the linearized sideslip angle, i.e. $\Delta\beta_f = \Delta\beta$.

Barometric Altimeter

Altimeters are sensors that measure altitude. There are two general types, passive signal reception and active signal transmission and reception. The primary passive altimeter is the barometric altimeter. The three types of active signal transmission and reception are active sonar, radar, and lidar altimeters. **Barometric altimeters**, also known as **pressure altimeters** measure the ambient pressure and use a pressure map to invert this to an estimated altitude, typically using the International Standard Atmosphere (ISA) model which linearly relates absolute temperature of the air, T , with altitude, h , at different atmospheric layers. For the pressure, P , the ISA and barometric altimeters generally use the hydrostatic balance and ideal gas law to get

$$h = \frac{RT}{g} \log \left(\frac{P_0}{P_h} \right) \quad (18.28)$$

where R is the specific gas constant for dry air ($287.058 \text{ J kg}^{-1} \text{ K}^{-1}$), T is the absolute temperature of the air, g is the acceleration due to gravity, P_h is the pressure at altitude h , and P_0 is the reference pressure at MSL.

Differences in temperature from the ISA model will also cause errors in the measured altitude as well as small errors due to assumptions about the acceleration due to gravity in the altimeter due to its variance with latitude and altitude. Barometric altimeters can exhibit errors of hundreds of feet due to sudden changes in air pressure, e.g. weather fronts, without any actual change in altitude. Modern aircraft use a barometric altimeter known as a **sensitive altimeter** which allows one to adjust the reference pressure at MSL which is supplied by local airports since MSL reference atmospheric pressure at a given location varies over time with temperature and the movement of pressure systems in the atmosphere.

ADS Errors

Pitot-static ADS faults can occur from a blockage of the air passage due to foreign objects. A **blocked pitot tube** is caused by objects in the atmospheric entering and obstructing the tube, e.g. ice, water, or insects. Thus, the FAA recommends pitot tubes be checked for obstructions prior to any flight and to prevent icing, all

pitot tubes for instrument-certified flight are equipped with a heating element. This blockage will only affect the airspeed measurement as a blocked pitot tube will measure an increase or decrease in airspeed when the aircraft climbs or descends, even though actual airspeed is constant due to the pitot pressure remaining constant while the static pressure is decreasing/increasing. However, a **blocked static port** typically only occurs due to icing. A blocked static port will cause the system to measure a constant altitude at which the static source became blocked, the measured rate of climb will be a constant value of zero, and the airspeed be measured as less or more than its true value as aircraft climbs or descends. This fault is much more dangerous than a blocked pitot tube because it affects *all* pitot-static differential pressures.

Inherent errors fall into several categories, each affecting different instruments. A **density error** is caused by variations of pressure and temperature in the atmosphere and affects the airspeed and altitude measurements. A **compressibility error** can arise because the ram air pressure will cause the air to compress inside the pitot tube. At standard sea level pressure altitude, this can be calibrated, but at higher altitudes, the compression may cause the sensor to measure greater than equivalent airspeed if a correction is not tabulated. These compressibility errors become significant at altitudes above 3,000 m and at airspeeds greater than 100 m/s. A **reversal error** is caused by a false static pressure measurement. This false measurement is typically caused by abnormally large changes in an aircraft's pitch. A large change in pitch will cause a momentary sensing of movement in the opposite direction. Reversal errors primarily affect the altitude and rate of climb measurements.

A **position error** is produced by the aircraft's static pressure being different from the air pressure of the ambient air mass far away from the aircraft due to the air flowing past the static source at a speed different from the aircraft's true airspeed. Position errors depend on several factors including airspeed, angle of attack, aircraft weight, acceleration, aircraft configuration, and in the case of rotorcraft, downwash. There are two categories of position errors: **fixed errors** and **variable errors**. Fixed errors are defined as specific to a particular model of aircraft, while variable errors are caused by external factors such as deformed panels obstructing the flow of air, or particular situations which may overstress the aircraft. A **lag error** is caused by the fact that any changes in the static or dynamic pressure outside the aircraft require a finite amount of time to affect the air in the tubing and the associated pressure gauges. Naturally, this type of error depends on the length and diameter of the tubing as well as the volume inside the gauges. Lag error is only significant around the time when the airspeed or altitude are changing rapidly and is not a concern for straight and level flight.

18.5 Fuel and Battery Monitoring Systems

Fuel Monitoring Systems

A **fuel monitoring system** estimates the fuel level of a vehicle. A typical monitoring system uses two sensors, a **fuel flow meter** and a **fuel tank level** as a simple system modeled as

$$\begin{aligned} \begin{bmatrix} f_k \\ b_k \end{bmatrix} &= \begin{bmatrix} 1 & A_{line}\Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} f_{k-1} \\ b_{k-1} \end{bmatrix} + \begin{bmatrix} -A_{line}\Delta t \\ 0 \end{bmatrix} u_{k-1} + \vec{w} \\ y_k &= [A_{tank}^{-1} \quad 0] \begin{bmatrix} f_k \\ b_k \end{bmatrix} + v \end{aligned} \tag{18.29}$$

where

- f_k is the fuel remaining in cm^3 ,
- b_k is the flow meter bias in cm,
- A_{line} is the cross-sectional area of the fuel line in cm^2 ,
- Δt is the sampling rate in s,
- u_{k-1} is the fuel flow meter measurement at time step $k - 1$ in cm/s ,
- \vec{w} is the zero-mean fuel flow meter measurement error in $[\text{cm}^3, \text{cm}]^T$ with covariance Q ,
- y_k is the fuel tank level measurement in cm,
- A_{tank} is the cross-sectional area of the fuel tank in cm^2 , and
- v is the fuel tank level measurement error in cm with variance R .

Note that in this model, the bias drift is modeled as a Gaussian random walk, but could be modeled as an $AR(1)$ process or a jump process in some similar sensors.

Battery Monitoring Systems

A model-based **battery monitoring system (BMS)** typically estimates three quantities, the **state-of-charge (SoC)** of the battery, the **state-of-health (SoH)**, and the **remaining useful life (RUL)**, using the voltage and current measurements at the nodes of the battery with three cascaded nonlinear Bayesian filters.

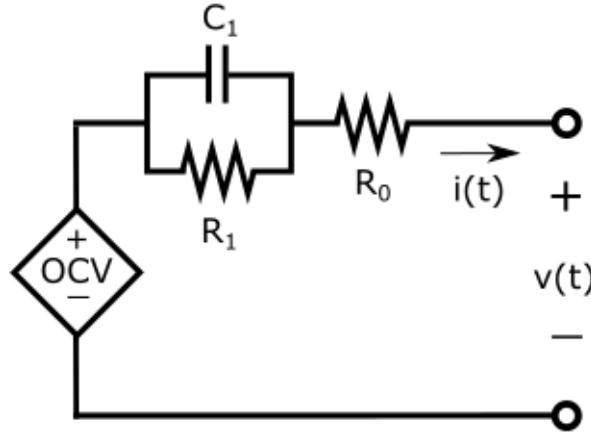
The SoC is defined as the percentage of current i remaining in the battery relative to the capacity, Q , i.e., as the

$$\dot{SoC} = -\frac{i(t)}{Q} \quad (18.30)$$

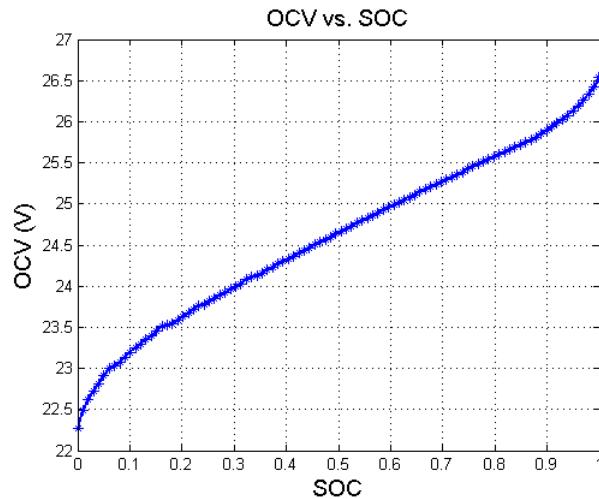
where one can use numerical integration to estimate the SoC if one can measure the current. Note that here $i(t)$ should be positive for a discharging battery. Note that the total capacity of the battery can be considered as all of the possible current usable in the battery, i.e.

$$Q = \int_0^\infty i(t) dt \quad (18.31)$$

To relate the voltage and current measurements to the SoC and Q , consider the following equivalent circuit diagram of a battery.



where R_0 is the internal resistance, $\tau_1 = R_1 C_1$ is one dynamic element of the battery (there may be multiple of these), $v(t)$ is the voltage measurement, $i(t)$ is the current measurement, and **OCV** is the **open circuit voltage (OCV)** of the battery. Empirically, it is known that the OCV is a nonlinear function of the SoC as shown in the following figure.



which remains relatively constant over the lifetime of the battery.

With these two sensors of current and voltage, the *SoC* estimation and one *RC* element allows one to form the following state-space model

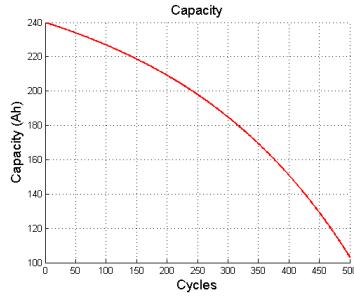
$$\begin{bmatrix} SoC_k \\ i_{1,k} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & \exp^{-\frac{\Delta t}{R_1 C_1}} \end{bmatrix} \begin{bmatrix} SoC_{k-1} \\ i_{1,k-1} \end{bmatrix} + \begin{bmatrix} -\frac{\Delta t}{Q} \\ 1 - \exp^{-\frac{\Delta t}{R_1 C_1}} \end{bmatrix} i(k-1) \quad (18.32)$$

$$v(k) = OCV(SoC_k) - R_0 i(k) - R_1 i_1(k)$$

The SoH is the current value of the capacity $Q(t)$ relative to the initial capacity, Q_0 , i.e.,

$$SoH = \frac{Q(t)}{Q_0} \quad (18.33)$$

which will slowly decrease over the lifetime of the battery as shown in the following figure.



which is typically modeled as an exponential decay with the number of cycles, ℓ , as a double exponential model

$$Q(\ell) = ae^{b\ell} + ce^{d\ell} \quad (18.34)$$

However, some batteries exhibit “burn-in” growth stage, a plateau stage, and then a decay stage which requires additional detection algorithms in estimation.

However, as $Q(\ell)$ can change $< 1\%$ during a single discharge cycle of the battery, estimating $Q(\ell)$ continuously with k during a discharge cycle is unnecessary and can actually lead to volatility of the estimate. Thus, cascading *SoH* estimation after *SoC* estimation improves performance. To setup a cascaded Bayes filter, step propagates the *SoC* at the beginning of a discharge cycle, SoC_{start} , through the *entire* ℓ^{th} discharge cycle with N_ℓ time steps with current capacity estimate and current measurements, i.e.

$$\hat{SoC}_{proj} = SoC_{start} - \frac{\sum_{k=0}^{N_\ell-1} i_k}{SoHQ_0} \quad (18.35)$$

Then, one can correct the Q estimate with the observed *SoC* estimate at the end of the cycle, SoC_{end} . This provides the stochastic state-space model with a nonlinear measurement model as

$$SoH_\ell = SoH_{\ell-1} + \vec{w}_{\ell-1} SoC_\ell = \frac{\sum_{k=0}^{N_\ell-1} i_k}{SoH_\ell Q_0} \quad (18.36)$$

Notably, this method relies heavily on accurate *SoC* estimates across the entire discharge cycles.

The RUL is defined as the number of discharge cycles remaining before one reaches a *SoH* that is unusable for the system employing the battery, e.g., 50-80%. By recursively estimating the model parameters of the exponential decay model, $\vec{x} = [a \ b \ c \ d]^T$, the RUL can be calculated by numerically solving the equation

$$SoH_{min} = \hat{a}e^{\hat{b}RUL} + \hat{c}e^{\hat{d}RUL} \quad (18.37)$$

18.6 Fault and Integrity Monitoring Systems

In the operation of multi-sensor systems, one or more sensors in the system may malfunction from its intended purpose, i.e. a sensor may experience a **fault**. In many designs, information systems often include **fault monitoring systems** to detect and isolate potential faulted sensor(s) from the multi-sensor data fusion algorithm, also known as **fault detection and isolation (FDI)** methods, which are either model-based or signal processing-based. Model-based FDI typically uses **stochastic state-space fault** models, i.e. for exteroceptive sensors as

$$\vec{y}_k = h_k(\vec{x}_k, \vec{v}_k, \vec{b}_k) \quad (18.38)$$

and proprioceptive sensors as

$$\vec{x}_k = f_{k-1}(\vec{x}_{k-1}, \vec{u}_{k-1}, \vec{w}_{k-1}, \vec{b}_{k-1}) \quad (18.39)$$

where \vec{b} is the **fault vector**, also known as the **bias vector**.

The fault vector may be constant or time-varying and deterministic or stochastic. Thus, one typically uses detection theory to formulate a detector and threshold based on the fault model(s) for \vec{b} and the type of data fusion algorithm.

Here, if one detects any faulted measurements and identifies the sensor associated with the faulted measurement, then one should only use the subset of fault-free measurements, thereby excluding or isolating any faulted measurements from the fusion algorithm, a process known as **fault detection and exclusion (FDE)** or **fault detection and isolation (FDI)**. A common method for model-based FDE/FDI is called **solution separation** or **N-model redundancy** whereby one evaluates the data fusion solution for the full set of measurements and cycles through all subsets of measurements based on all fault models which notably may have single- or multiple-fault modes. Then, a detector, D , is formed from the solution based on each subset and the solution based on the full set which is then compared to some threshold, τ . If $D > \tau$, then that fault-mode is declared and those faulted measurements are removed from the final data fusion output. However, these types of multi-sensor systems require

For constant, single-faults, the solution separation can be shown for OLS parameter estimation as follows. Consider n_y measurements linearly related to the parameter, i.e.

$$\vec{y} = X\vec{\beta} + \vec{v} + \vec{b} \quad (18.40)$$

where $n_y > n_\beta$, $v_i \sim \mathcal{N}(0, \sigma_v^2)$, $b_i = 0$ when the i^{th} sensor measurement y_i is fault-free and $b_i = \mu_i$ when the i^{th} sensor measurement y_i is faulted. Recall the OLS solution is

$$\hat{\vec{\beta}} = (X^T X)^{-1} X^T \vec{y} \quad (18.41)$$

and the **measurement residual** is a Gaussian distributed random vector given by

$$\vec{r} = \vec{y} - \hat{\vec{y}} = \vec{y} - X\hat{\vec{\beta}} \quad (18.42)$$

where one can show

$$\vec{r} = (I - X(X^T X)^{-1} X^T)(\vec{v} + \vec{b}) \quad (18.43)$$

Then, one can define the detector as

$$D = \frac{1}{\sigma_v^2} \vec{r}^T \vec{r} \quad (18.44)$$

$$D = \frac{1}{\sigma_v^2} \left(r_1^2 + \cdots + r_{n_y}^2 \right) \quad (18.45)$$

For fault-free measurements, i.e.

$$\mathbb{E} [\vec{r}] = \vec{0} \quad (18.46)$$

it can be shown that $D \sim \chi^2(n_y - n_\beta)$, i.e. a centralized chi-squared with $n_y - n_\beta$ degrees-of-freedom. For single-fault measurements, i.e.

$$\mathbb{E} [\vec{r}] = [0 \ \cdots \ \mu_i \ \cdots \ 0]^T \quad (18.47)$$

it can be shown that $D \sim \chi^2(n_y - n_\beta, \frac{\mu_i^2}{\sigma_v^2})$, i.e. a noncentralized chi-squared with $n_y - n_\beta$ degrees-of-freedom and a $\frac{\mu_i^2}{\sigma_v^2}$ non-centrality parameter. Thus, for a given expected fault magnitude, μ_i and P_{FA} , one can formulate a threshold, τ , and probability of missed detection of the fault, P_{MD} , based on the χ^2 distribution models.

Integrity Monitoring System

Multi-sensor systems are often used in safety-critical operations for which it is imperative one has quantified the fault-free and faulted performance using risk analysis, namely, integrity risk and continuity risk. In this context, a **integrity monitoring system**, also known as a **integrity monitor**, assesses the safety-critical outputs for an potential **hazardously misleading information (HMI)** which is defined for specific types of safety-critical operations. **Integrity risk**, also known as the **probability of hazardously misleading information (HMI)**, P_{HMI} , is probability of a hazardously large output error occurring *without* the integrity monitor issuing an alert. As the risk of a large error can never practically be completely eliminated, it is necessary to accept a finite, albeit very small, risk of HMI. Integrity risk is divided into two sub-categories, *faulted* and *fault-free*. Faulted integrity risk is the probability that the integrity monitor will fail to issue an alert when it should due to a system fault.

Thus, one can model the integrity risk, P_{HMI} , as

$$P_{HMI} = P_{HMI|0} + P_{HMI|F}P_F + P_{MI|D}P_D \quad (18.48)$$

where $P_{HMI|0}$ is the **probability of hazardously misleading information (HMI)** with a fault-free system, $P_{HMI|F}$ is the **probability of hazardously misleading information (HMI)** with a faulted system, P_F is the probability of a fault, P_D is the **probability of fault detection**, and $P_{MI|D}$ is the **probability of a mis-identification** of the fault which leads to HMI *given* a system fault was detected. Furthermore, the first two terms can be grouped together to obtain

$$P_{MD} = P_{HMI|0} + P_{HMI|D}P_D \quad (18.49)$$

where P_{MD} is the **probability of missed detection** of HMI.

Continuity risk is the probability that an integrity monitor alert interrupts a safety-critical operation reliant on the information system. Integrity monitor alerts due to detected failures cause the failed measurements to be excluded from use, an action that may in turn reduce the information accuracy to a level at which the current operation is no longer safe to continue. In addition, integrity monitors may trigger false alerts. Thus, one can model the continuity risk, R_C , as

$$R_C = P_{FA} + P_{NI}P_D \quad (18.50)$$

where P_{FA} is the **probability of false alarm** of a HMI alert when HMI has not actually occurred and P_{NI} is the **probability of a non-isolable fault** for which the integrity monitor declares an HMI alert as it could not isolate the fault *given* one has detected a fault which could result in HMI. Continuity requirements in effect dictate integrity monitor thresholds and determine integrity monitor sensitivity. They also set maximum allowable fault probabilities for the underlying sensors.

As shown in the equations above, integrity monitors use FDI methods to determine the two terms within the integrity and continuity risks. In this context, an integrity monitor compares the safety-critical outputs to pre-defined **alert limits (ALs)**, i.e. the maximum allowable operational error beyond which the integrity monitor should issue a HMI alert. In this case, the “true” integrity risk, P_{HMI} , can be restated as the probability that, at any moment, the true output error exceeds the AL. However, as one does not typically know the exact output error statistics, e.g. the measurement noise variance σ_v^2 , integrity monitors compute **protection levels (PLs)** using **error overbounds** of the system output. Overbounds model the probabilities of exceedance at all error levels as smaller than or equal to the true integrity risk of the system’s operation. Thus, protection levels are used to ensure with a very high probability that the provided information will not result in a hazardous situation with a very high confidence level. With these protection levels and alert limits, an integrity monitor uses a detector D to detect a system fault based on some threshold, τ , and the integrity monitor can bound the integrity risk as

$$\Pr(PL > AL | \mathcal{H}_0) + \Pr(PL > AL \cup D < \tau | \mathcal{H}_F) \leq P_{HMI} \quad (18.51)$$

where \mathcal{H}_0 is the fault-free system hypothesis and \mathcal{H}_F is the faulted system hypothesis.

As an example of an integrity monitoring system, consider one given an alert limit, AL , for the parameter estimate, $\vec{\beta}$ of a multi-sensor information system. Then, a single-fault **solution separation method** for integrity monitor can be outlined as follows.

1. Check measurement dimension n_y exceeds the parameter dimension n_β
2. Determine suitable detection threshold, τ
3. Compute parameter estimate using all n_y measurements
4. FDI for $k = 1, \dots, n_y$ measurements:
 - Compute parameter estimate with declared “fault-free” and unevaluated measurements excluding k
 - Compute fault detector, D_α , with declared “fault-free” and unevaluated measurements excluding k
 - Declare measurement k “fault-free” if $D_\alpha < \tau$
5. Compute protection levels based on “fault-free” measurements and error overbounds: PL
6. Declare *system available* if $PL \leq AL$

For many applications, these types of requirements on any potentially hazardously misleading information must often be computed by the multi-sensor systems during operation. Thus, a large part of *safety-critical* multi-sensor systems is the characterization of statistics of the estimate errors, i.e. the uncertainty quantification of the integrity risk. The performance of an integrity monitor is typically denoted by the **availability**, i.e. the fraction of the time that integrity and continuity requirements can be satisfied by the information system. To assess whether performance requirements are met, it is necessary to conduct an available test

which confirms that integrity, and potentially continuity and accuracy as well, is met for both fault-free and faulted conditions.

Error Overbounding

Unfortunately, one does not always have accurate sensor error models used in model-based multi-sensor fusion, i.e. the probabilistic models may be difficult to exactly quantify. This leads to the concept of **overbounding** which models the probability of an error as always greater than or equal to the true probability. Mathematically, if the error is defined as a random negative or positive “distance” away from some error-free value \tilde{x} , one can model this error as the random variable X and $B(x)$ is an **overbound of X** about \tilde{x} if

$$\begin{aligned} B(x) &\geq \Pr(X \leq x) \quad \forall x < \tilde{x} \\ B(x) &\geq \Pr(X > x) \quad \forall x > \tilde{x}. \end{aligned} \quad (18.52)$$

i.e. the overbound provides upper bounds for the probabilities of observing any negative error, i.e. $x < \tilde{x}$, and any positive error, i.e. $x > \tilde{x}$.

This separation of positive and negative error bounding is useful for the fundamental definition of overbounding since X may contain asymmetry about \tilde{x} . Furthermore, this overbound definition can be rewritten in terms of the *unknown* CDF of X i.e.

$$\begin{aligned} B(x) &\geq F_X(x) \quad \forall x < \tilde{x} \\ B(x) &\leq F_X(x) \quad \forall x > \tilde{x}. \end{aligned} \quad (18.53)$$

It should be noted that this formulation puts no requirement on the actual value of $B(x)$ at $x = \tilde{x}$, which may be relevant for mixed random variables, but has no effect on continuous random variables. This section defines three types of univariate error overbounds for multi-sensor information systems. These cases are shown graphically along with the depiction of a general overbound. If $F(x) = 0 \quad \forall x \leq \tilde{x}$, then $B(x)$ is the **one-sided overbound** of X if

$$B(x) \leq F_X(x) \quad \forall x > \tilde{x}. \quad (18.54)$$

If $F(\tilde{x}) = 0.5$ where \tilde{x} = median of x , then $B(x)$ is the **two-sided overbound** of X if

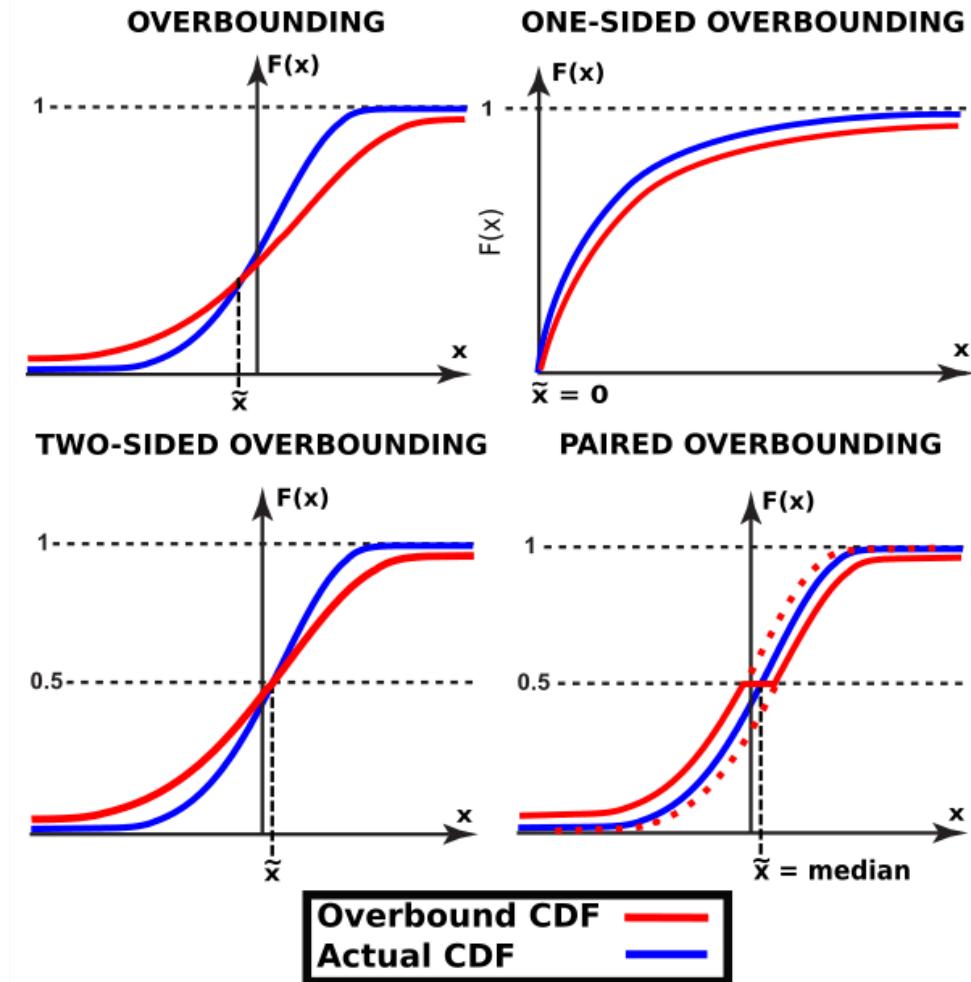
$$\begin{aligned} B(x) &\geq F_X(x) \quad \forall F_X(x) < 0.5 \\ B(x) &\leq F_X(x) \quad \forall F_X(x) > 0.5. \end{aligned} \quad (18.55)$$

From this definition of the two-sided overbound, a common construction called paired overbounding. This approach can also be considered as a natural extension of one-sided overbounding to the two-sided case, where two separate models are used, one for each side of $F(x) = 0.5$. Mathematically, $B(x)$ is called the **paired overbound** of X if

$$B(x) = \begin{cases} B_L(x) & \forall x : B_L(x) < 0.5 \\ 0.5 & \text{otherwise} \\ B_R(x) & \forall x : B_R(x) > 0.5 \end{cases} \quad (18.56)$$

where $B_L(x)$ and $B_R(x)$ satisfy

$$\begin{aligned} B_L(x) &\leq F_X(x) \quad \forall x \\ B_R(x) &\geq F_X(x) \quad \forall x. \end{aligned} \quad (18.57)$$



Although the mathematical formulations for the preceding overbounding definitions hold in principle, the fact that the true random variable remains *unknown* requires one estimate the overbound model of the error. Thus, one typically uses error data to form the **empirical distribution function (EDF)** of X , $\hat{F}_X(x)$, given by

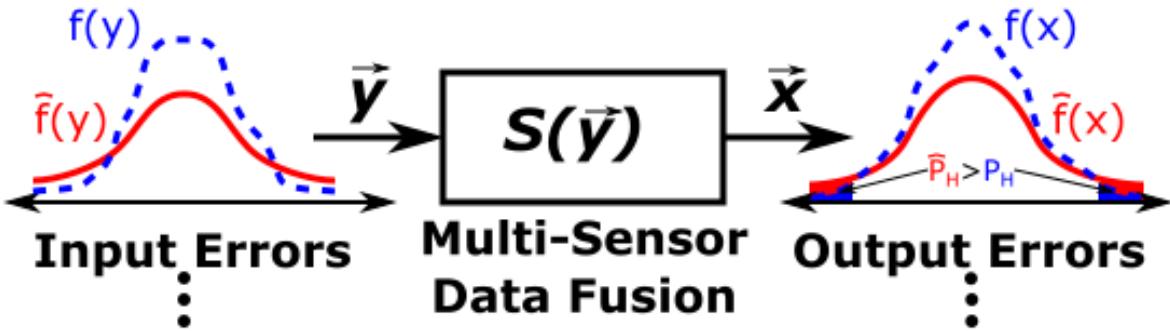
$$\hat{F}_X(x) = \begin{cases} 0, & \forall x < x_{(1)} \\ i/n, & \forall x_{(i)} \leq x < x_{(i+1)} \quad i = 1, \dots, n-1 \\ 1, & \forall x \geq x_{(n)} \end{cases} \quad (18.58)$$

and $x_{(1)} \leq x_{(2)} \leq \dots \leq x_{(n)}$ is the ordered magnitudes of the samples x_1, \dots, x_n .

In addition, one typically desires to be able to compute overbounds for the output errors of multi-sensor information systems. This can be done by mapping the input error overbounds through the data fusion algorithm. When the algorithm is linear, the mapping can be computed by convolving the input error distributions. It has been proven that the convolution of one- or two-sided overbounds produce an overbound on the convolution of the true error distributions if they are unimodal, symmetric, and zero-mean.

Furthermore, it has been shown that paired overbounds can remove these limitations by convolving each pair of overbounds separately. This convolution approach to output overbounding has also led to the Gaussian distribution becoming by far the most popular overbound model primarily because convolutions of Gaussian random variables remain Gaussian random variables. However, if the overbounded distribution exhibits heavy-tailedness, i.e. it decays slower than a Gaussian distribution, then the input and output errors cannot be truly overbounded by a Gaussian overbound model.

The following figure displays the concept of applying the overbounding method for multi-sensor information systems by using conservative overbounds for calculating the probabilities of hazardously misleading information, P_{HMI} . Such a system with a linear mapping can be mathematically represented as



$$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} = S \begin{pmatrix} y_1 \\ \vdots \\ y_m \end{pmatrix} = S \begin{bmatrix} y_1 \\ \vdots \\ y_m \end{bmatrix} = \begin{bmatrix} \sum_{j=1:m} S_{1,j} y_j \\ \vdots \\ \sum_{j=1:m} S_{n,j} y_j \end{bmatrix} \quad (18.59)$$

where $[y_1 \dots y_m]^T$ and $[x_1 \dots x_n]^T$ are the input and output random vectors of lengths n and m , respectively, and S is the $n \times m$ linear transformation matrix with $S_{i,j}$ representing the element of the i^{th} row and j^{th} column. Next, since the original inputs are first scaled, the scaled inputs can be computed by the formula for a function $X = S(Y)$ of random variable, Y :

$$f_X(x) = f_Y(S^{-1}(y)) \left| \frac{dS^{-1}(y)}{dy} \right| \quad (18.60)$$

where if $S(Y) = SY$, then

$$f_X(y) = \frac{1}{|S|} f_Y \left(\frac{y}{S} \right) \quad (18.61)$$

Then, if the inputs are independent, the output elements $x_1, \dots, x_i, \dots, x_n$ can be written as successive $m-1$ convolutions of the PDFs of inputs $y_1, \dots, y_j, \dots, y_m$ where $f_k(x)$ denotes the PDF for the k^{th} convolution which for $1 < k \leq m-1$ is computed by

$$f_k(u) = \int_{-\infty}^{\infty} \frac{1}{|S_{i,m-k}|} f_{y_{m-k}} \left(\frac{t_k}{S_{i,m-k}} \right) f_{k-1}(u - t_k) dt_k \quad (18.62)$$

and for $k = 1$ is given as

$$f_1(u) = \int_{-\infty}^{\infty} \frac{1}{|S_{i,m-1}|} f_{y_{m-1}}\left(\frac{t_1}{S_{i,m-1}}\right) \frac{1}{|S_{i,m}|} f_{y_m}\left(\frac{u - t_1}{S_{i,m}}\right) dt_1 \quad (18.63)$$

where $f_{m-1}(\cdot) = f_{x_i}(\cdot)$ is the PDF of output x_i and $f_{y_j}(\cdot)$ is the PDF of input y_j .

Now suppose that y_1, \dots, y_m are left overbounded by CDFs $B_{L,y_1}(y), \dots, B_{L,y_m}(y)$ and right overbounded by CDFs $B_{R,y_1}(y), \dots, B_{R,y_m}(y)$, respectively, then performing the same successive convolutions for the overbounding PDFs and integrating the resulting PDFs to CDFs produces left overbounds, $B_{L,x_1}(x), \dots, B_{L,x_n}(x)$ and right overbounds $B_{R,x_1}(x), \dots, B_{R,x_n}(x)$ for x_1, \dots, x_n , respectively. When the overbounds used for the inputs are Gaussians, then equations 18.62 and 18.63 do not need to be computed explicitly, but only the means and variances must be computed using the algebraic formulas that follow. This is the primary reason why Gaussian overbounds are simpler to handle for real-time integrity computations. The mean formula is

$$\mu_{x_i} = \sum_{j=1:m} S_{i,j} \mu_{y_j} \quad (18.64)$$

and the variance formula is

$$\sigma_{x_i}^2 = \sum_{j=1:m} S_{i,j}^2 \sigma_{y_j}^2 \quad (18.65)$$

where μ_{x_i} and μ_{y_j} are the means of the distributions of x_i and y_j , respectively, and $\sigma_{x_i}^2$ and $\sigma_{y_j}^2$ are the variances of the distributions of x_i and y_j , respectively. However, for input overbounds that are not Gaussian, then equations 18.62 and 18.63 must be computed numerically.

18.7 Clocks and Timing Systems

Timing systems provide a time output using sensors known as **clocks**, also known as **timepieces** or **chronometers**, which are studied as part of **horology**. Throughout history, different devices have been used as clocks including sundials, hourglasses, and water clocks. However, modern clocks use harmonic oscillators, also known as a **resonator**, which is a physical object that vibrates or oscillates at a particular *known* frequency. These include pendulums, tuning forks, quartz crystals, i.e. quartz clocks, or atomic vibrations from electrons emitting microwaves, i.e. **atomic clocks**. In most information systems, either quartz clocks or atomic clocks are used.

The piezoelectric properties of crystalline quartz allows crystal oscillators to be used in **quartz clocks**. The National Bureau of Standards based the time standard of the United States on quartz clocks from late 1929 until the 1960s when it changed to atomic clocks. However, their inherent accuracy and low cost of production resulted in the subsequent proliferation of quartz clocks. Currently, **atomic clocks** are the most accurate clocks in existence. They are considerably more accurate than quartz clocks as they can be accurate to within a few seconds over trillions of years. In the 1930s the development of magnetic resonance created practical methods for developing a prototype ammonia maser device in 1949 which was initially less accurate than existing quartz clocks. The first accurate atomic clock based on a certain transition of the caesium-133 atom was built in 1955. As of 2013, the most accurate atomic clocks are ytterbium clocks.

The development of atomic clocks has led to many scientific and technological advances, e.g. navigation satellite systems, long-baseline interferometry in radioastronomy, and the Internet, all of which critically depend on frequency and time standards. In these applications, atomic clocks are installed at broadcasting stations with radio transmitters which transmit a very precise carrier frequency used as a time signal. Many governments operate these transmitters for time-keeping purposes. A **radio clock** is a clock that synchronizes itself by means of these radio time signals received by a radio receiver. Consumer-grade receivers solely rely on the amplitude-modulated time signals and use narrow-band receivers, 10 Hz bandwidth, with small ferrite loopstick antennas and circuits with digital signal processing delays. Consumer-grade radio clocks are expected to determine the beginning of a second with an uncertainty of ± 0.1 second. Instrument-grade time receivers provide higher accuracy and incur a transit delay of approximately 1 ms for every 300 km of distance from the radio transmitter.

Time Standards

A **time standard** is a specification for measuring time, either the rate at which time passes or specific points in time or both. Historically, time standards were often based on the Earth's rotational period as it was assumed that the Earth's daily rotational rate was constant which is false. These were replaced for astronomical use from 1952 onwards by an ephemeris time standard based on the Earth's orbital period and on the motion of the Moon. However, the invention of atomic clocks has led to the replacement of astronomical time standards by newer time standards based on atomic time.

Ephemeris time (ET) is a former standard astronomical time scale adopted in 1952 and superseded during the 1970s. This time scale was proposed to overcome the disadvantages of irregularly fluctuating mean solar time and to define a uniform time based on Newtonian theory. ET was a first application of the concept of a dynamical time scale, in which the time and time scale are defined implicitly, inferred from the observed position of an astronomical object via the dynamical theory of its motion.

Terrestrial time (TT) is a modern astronomical time standard defined by the International Astronomical Union and defines a coordinate time scale at the Earth's surface primarily for time-measurements of astronomical observations made from the Earth's surface. As TT shares the original purpose for which ET was designed, to be free of the irregularities in the rotation of Earth, it has succeeded the use of ET. It is a theoretical ideal, and clocks can only approximate it. The unit of TT is the **SI second**, the definition of which is based currently on the caesium atomic clock, though TT is not itself defined by atomic clocks.

International Atomic Time (TAI) is the primary international time standard from which other time standards are calculated. TAI is kept by the International Bureau of Weights and Measures (BIPM) and is based on the combined input of many atomic clocks around the world, each corrected for environmental and relativistic effects, both gravitational and speed. Of note, because of the historical difference between TAI and ET when TT was introduced, TT is approximately 32.184 seconds ahead of TAI.

Universal Time (UT) is the mean solar time at 0° longitude. However, as precise measurements of the sun are difficult, UT is computed from determining the positions of distant quasars using very long baseline interferometry (VLBI), satellite laser ranging (SLR), and global navigation satellite systems (GNSS). These observations allow the determination of a measure of the **Earth rotation angle (ERA)**, i.e. Earth's angle with respect to the ICRF, neglecting some small adjustments. The principal form of UT, UT1 is required to follow the following relationship.

$$ERA = 2\pi(0.7790572732640 + 1.00273781191135448(JD - 2451545.0)) \text{ radians} \quad (18.66)$$

where ***JD*** is the **Julian date (JD)**, i.e. the Julian day number plus the fraction of a day since the preceding noon in UT1 where the **Julian day number (JDN)** is the continuous count of days since the beginning of the Julian period, i.e. noon Universal Time on January 1, 4713 BC, by the Julian calendar and November 24, 4714 BC, by the Gregorian calendar.

Coordinated Universal Time (UTC) is an atomic time scale designed to approximate UT. UTC differs from TAI by an integral number of seconds and is kept within 0.9 second of UT1 by the introduction of one-second steps to UTC, i.e. the “**leap second**.” To date these leap-seconds and the difference, TAI-UTC, have always been positive.

Standard time, also known as **civil time** in a region deviates a fixed, round amount, usually a whole number of hours, from UTC. The offset is chosen such that a new day starts approximately while the sun is crossing the nadir meridian. It may also change twice a year by a round amount, usually one hour, e.g. **daylight saving time (DST)**.

Timing systems within the context of information systems often use their own time standard which are typically related in some defined way to UTC. One common standard widely used in operating systems and file formats is **posix time**, also known as **unix time**, defined as the number of seconds that have elapsed since 00:00:00 UTC on January 1, 1970, excluding leap seconds. Posix time is nonlinear as a UTC leap second has the same Unix time as the second before it or after it, so that every day is treated as if it contains exactly 86400 seconds with no seconds added to or subtracted from the day as a result of positive or negative leap seconds.

Clock Stability and Allan Variance

Quartz and atomic oscillators of clocks have a phase noise consisting of white noise and **flicker frequency noise** which is a divergent noise. Thus, all clocks exhibit **clock drift**. Furthermore, as this noise is divergent, one cannot compute a standard deviation. Thus, early efforts in analyzing the stability included both theoretical analysis and practical measurements. However, as various methods of measurements did not agree with each other, the key aspect of repeatability of a measurement could not be achieved, limiting the possibility to compare clock sources and make meaningful specifications to require from suppliers. Essentially all forms of scientific and commercial uses were then limited to dedicated measurements, which hopefully would capture the need for that application.

To address these problems, David Allan introduced the ***M*-sample variance** and, indirectly, the **2-sample variance**, also known as the **Allan variance**. The *M*-sample variance is defined as

$$\sigma_y^2(M, T, \tau) = \frac{1}{M-1} \left(\sum_{k=0}^{M-1} \bar{y}_k^2 - \frac{1}{M} \left(\sum_{k=0}^{M-1} \bar{y}_k \right)^2 \right) \quad (18.67)$$

where *M* is the number of frequency samples used in the variance, *T* is the time between adjacent start events, τ is the observation period from start to stop events, and \bar{y}_k is the *k*th **average fractional frequency** defined

$$\bar{y}_k = \frac{x(kT + \tau) - x(kT)}{\tau} \quad (18.68)$$

where $x(t)$ is the clock reading (in seconds) measured at *t*. It should be noted that the *M*-sample allows the use of **dead-time** by letting *T* be different from τ . Allan proved all *M*-sample variances were comparable

and did not converge for large M , thus the Allan variance has become the preferred variance. The Allan variance is defined as

$$\sigma_y^2(\tau) = \mathbb{E} [\sigma_y^2(2, \tau, \tau)] \quad (18.69)$$

$$\sigma_y^2(\tau) = \frac{1}{2} \mathbb{E} [(\bar{y}_{k+1} - \bar{y}_k)^2] \quad (18.70)$$

or

$$\sigma_y^2(\tau) = \frac{1}{2\tau^2} \mathbb{E} [(x(k\tau + 2\tau) - 2x(k\tau + \tau) + x(k\tau))^2] \quad (18.71)$$

Lastly, the **Allan deviation** is given by

$$\sigma_y(\tau) = \sqrt{\sigma_y^2(\tau)} \quad (18.72)$$

The Allan variance as one half of the time average of the squares of the differences between successive samples of the frequency deviation over the sampling period which is a free parameter. Therefore, it is a function of the distribution being measured and is displayed as a graph rather than a single number. While the Allan variance does not completely allow all types of noise to be distinguished, it provides a means to meaningfully separate many noise-forms for time-series of phase or frequency measurements between two or more oscillators. However, the Allan deviation is the preferred graphical measure as it gives the relative amplitude stability. A low Allan deviation indicates a clock has good stability over the measured period. An Allan deviation of 1.3×10^{-9} at an observation period of 1 s, i.e. $\tau = 1$ s, should be interpreted as there being an instability in frequency between two observations 1 second apart with a relative **root mean square (RMS)** value of 1.3×10^{-9} . For a 10 MHz clock, this would be equivalent to 13 mHz RMS movement. Ytterbium clocks are stable to within 2×10^{-18} in terms of Allan deviation. Though these definitions are given in terms of expectation, in practice, one must use estimate the Allan variance for which various statistical estimators have been proposed.

Clock Networks and Synchronization

A **clock network** is a set of synchronized clocks designed to always show exactly the same time by communicating with each other, e.g. **wireless sensor networks (WSN)** in multi-sensor data fusion. Each clock, $C_i(t)$, should be configured such that

$$C_i(t) = t \quad (18.73)$$

where t is the reference time. However, even when initially set accurately, real clocks will differ after some amount of time due to imperfections in the clock oscillator, i.e. clock drift. This clock drift can be modeled as

$$C_i(t) = \omega t + \theta \quad (18.74)$$

where ω is the **clock skew**, i.e. frequency difference, and θ is the **clock offset**, i.e. phase difference. Over time, these clock parameters vary due to external effects, e.g. temperature, atmospheric pressure, voltage changes, and hardware aging. Thus, the network has to perform periodic clock synchronization routines to adjust the estimated clock skew and offset and typically uses one of three schemes to perform clock synchronization: sender-receiver, receiver-receiver, or one-way sender.

A one-way sender strategy uses a master clock to broadcast its timing information, e.g. the **time of transmission (TOT)**, to slave clocks which also record the arrival times of the broadcast message, i.e. the

time of arrival (TOA). Traditionally, one models the master clock as the reference time, i.e.

$$C_M(t) = t \quad (18.75)$$

and a slave clock as having a relative clock skew and offset error terms

$$C_S(t) = \delta\omega t + \delta\theta \quad (18.76)$$

where the primary problem for each slave is to track the reference time t from the master clock given the TOT packets from the master. In modern applications, one typically uses digital clocks which use discrete clock ticks, i.e. an integer count of the periodic clock signal with some known period or **timebase**.

In this case, assume the master clock transmits a TOT packet with timebase T_M . This is received by the slave clock at time step k as $T_{S,k}$. With this model one can define the measurement model as

$$T_{S,k} \approx \delta\omega T_M + \delta\theta + \epsilon_k \quad (18.77)$$

where ϵ_k is the random variable delay in the transmission. Thus, clock tracking by each slave is implemented with the state dynamics taking three inputs, the slave clock TOA and the master clock's TOT with its timebase.

Digital phase-locked loops (DPLL) and Kalman filters have been widely used for clock synchronization.

Consider the following estimation of a phase and frequency offset for the master clock signal, i.e.

$$s_M(t) = A \cos \phi_M = A \cos(\omega_M t + \theta_M) \quad (18.78)$$

which is assumed to have some known values for ω_M and θ_M . A phase detector replicates this model and compares this to the slave clock signal, i.e.

$$s_S(t) = A \cos \phi_S = A \cos(\omega_S t + \theta_S) \quad (18.79)$$

which can be used to form the **loop phase error**, $\delta\phi$ at time step k as

$$\delta\phi_k = \phi_M - \phi_S = \delta\omega t + \delta\theta \quad (18.80)$$

In tracking mode, a DPLL $D(z)$ uses this to output a corrected sequence y_k

$$y_k = D(z)\delta\phi_k \quad (18.81)$$

where a proportional-integral DPLL

$$D(z) = K_p + K_i(1 - z^{-1})^{-1} \quad (18.82)$$

results in

$$y_k = K_p \delta\phi_k + K_i \sum_{j=0}^k \delta\phi_j \quad (18.83)$$

Then, the corrected sequence, y_k , is used to control the next period of the sample, T_{k+1} , through the adjustment

$$T_{k+1} = T_M - y_k \quad (18.84)$$

where one should adjust T_k until the loop is in the locked state with clock period T_k that exactly matches $t_{k+1} - t_k$. Furthermore, the reference sampling time for each packet can be obtained through

$$\hat{t}_{k+1} = \hat{t}_k + T_M - K_p \delta\phi_k - K_i \sum_{j=0}^k \delta\phi_j \quad (18.85)$$

Attitude Determination Systems

19.1 Introduction to Attitude Determination Systems

Notably, for spacecraft, many different features can be tracked including the Earth, the sun, the moon, and the stars. A **star tracker** is a sensor which uses optical photocells or cameras to sense the magnitude of brightness and spectral type to identify and measure the relative bearing vectors to specific identifiable stars, known as **navigational stars**. Thus, if one has a **star map** defined for the bearing vectors to identifiable stars in a known reference frame, e.g. the ICRF, ECI, or ECEF frame, then one can use these bearing vectors in the body-fixed frame to solve for the attitude of the spacecraft in the body-fixed frame relative to the LVLH frame by first transforming the reference bearing vectors to the LVLH or navigation frame using the estimated position of the vehicle first which couples the positioning and attitude determination problems.

Direct Roll and Pitch Determination

For roll ϕ and pitch θ Euler angle determination of the LVLH-to-body-fixed frame rotation, one uses infrared, optical, or inertial sensors for aerospace vehicles to obtain a **nadir** bearing measurement, \vec{b}_L , defined as straight down in the LVLH (or NED navigation frame) as

$$\vec{b}_L = \begin{bmatrix} 0 \\ 0 \\ \|\vec{b}_B\|_2 \end{bmatrix} \quad (19.1)$$

which can be related to the DCM by

$$\vec{b}_B = \begin{bmatrix} b_{B,1} \\ b_{B,2} \\ b_{B,3} \end{bmatrix} = C_{B \leftarrow L} \vec{b}_L \quad (19.2)$$

Thus, one has

$$\begin{bmatrix} b_{B,1} \\ b_{B,2} \\ b_{B,3} \end{bmatrix} = \|\vec{b}_B\|_2 \begin{bmatrix} -\sin \theta \\ \sin \phi \cos \theta \\ \cos \phi \cos \theta \end{bmatrix} \quad (19.3)$$

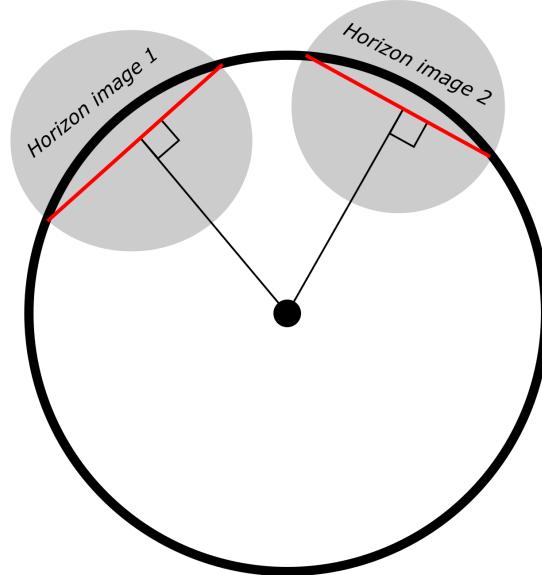
which implies one can estimate the pitch as

$$\hat{\theta} = \sin^{-1} \left(-\frac{b_{B,1}}{\|\vec{b}_B\|_2} \right) \quad (19.4)$$

and the roll as

$$\hat{\phi} = \tan^{-1} \left(\frac{b_{B,2}}{b_{B,3}} \right) \quad (19.5)$$

For spacecraft, infrared (IR) sensors can be used to sense the direction to the warmest source in the area of operation for satellites. If the source is the Earth's atmosphere, one has an **Earth sensor**. From the IR image, the centroid of the **Earth disk**, i.e. the 2D projection of the Earth on the image, must be rotated from the camera image frame to the body-fixed frame to construct the nadir vector measurement to the center of the Earth, i.e. an Earth sensor provides \vec{b}_B directly. Notably, at lower altitude orbits where the Earth disk is not completely contained with the sensor's field-of-view (FOV), the infrared imaging sensor is often referred to as an **Earth horizon sensor** from which the centroid must be estimated using the curvature of the horizon line, e.g. the intersection of the perpendiculars from two chords as shown.



For aircraft, an **accelerometer** is a sensor that directly measures the acceleration of a body in its own instantaneous rest frame. Conceptually, an accelerometer is a **proof mass**, also known as a **seismic mass**, on a spring. For planetary-based accelerometers, the rest frame is the Earth so the gravity force is not included.

When the accelerometer experiences an acceleration, the proof mass is deflected to the point that the spring accelerates the mass at the same speed as the casing. The deflection of the proof mass from its neutral position or the compression of the spring provides an acceleration measurement. However, the mass-spring system is also damped so that additional oscillations of the system do not affect the measurement output. Thus, all accelerometers have a frequency response and are designed to provide the required sensitivity and maximum expected acceleration. There are many different designs for converting this mass deflection or spring compression into a measurable electric signal including piezoelectric, piezoresistive, capacitive, and thermal.

A three-axis accelerometer measures the specific force experienced by the vehicle in the body-fixed frame, i.e.

$$\vec{f}_B = \vec{a}_B - \vec{g}_B \quad (19.6)$$

where \vec{a}_B is the total acceleration and \vec{g} is the gravitational acceleration where nominally, the gravity acceleration vector points in the nadir direction, i.e. straight down

$$\vec{g}_L = \begin{bmatrix} 0 \\ 0 \\ \|g_L\|_2 \end{bmatrix} \quad (19.7)$$

Thus, for an accelerometer, one has

$$\vec{b}_B = \vec{a}_B - \vec{f}_B \quad (19.8)$$

However, this requires one is able to independently calculate the acceleration of the vehicle or allow it as an unknown bias in the measurement. Thus, the quality of these estimates depends on the acceleration experienced by the vehicle which can be quite large. Thus, often accelerometer measurements for attitude are low-pass filtered to remove any high frequency accelerations.

Direct Yaw Determination

For a direct yaw solution for aerospace vehicles operating around Earth, one can use a magnetometer. A **magnetometer** is a sensor for measuring the strength and/or the direction of magnetic fields and can be based on a variety of techniques. The simplest is the use of a **magnetic compass** which is a permanently magnetized needle that is mounted so that it can pivot in the horizontal plane. With no interference, e.g. gravity, the needle will align itself exactly along the *local* magnetic field vector. Magnetic compasses typically have a response time less than 1 μ s and often can be sampled up to 1000 Hz. Many magnetic compasses provide accurate measurements within 1° of the magnetic field direction, for which the underlying compass must reliably resolve 0.1° variations.

Excepting local variations in magnetic materials, a magnetic compass nominally points in the direction of **magnetic north**, i.e. the direction of the *horizontal* component of the Earth's magnetic field relative to the Earth's ellipsoid at a given location. This direction is only approximately **geographic north**, also known as **true north**, i.e. the point about which the Earth rotates, and near the poles becomes a poor approximation. This error between the direction of local magnetic north and geographic north is called the **magnetic variation**, also known as the **magnetic declination** angle. The Earth's magnetic field suffers from localized variations which can also change over time. These variations are especially noticeable at high or low latitudes where one can measure a significantly large magnetic variation. Furthermore, the strength along these field lines vary based on location, thus, calibration is necessary for useful measurements.

A **three-axis magnetometer (TAM)** provides measurements of the Earth's magnetic *vector* coordinates and is often compensated for any dip errors with an additional sensor that estimates the pitch and roll angles. This algorithm called **tilt compensation**. However, a TAM is still susceptible to errors. In particular, the Biot-Savart Law states that ferromagnetic parts, and/or magnetic effects induced by electrical circuits will impact the TAM measurement as well. The presence of any additional constant magnetic fields will form the **hard-iron variation** and the presence of an additional induced magnetic field *dependent* on the TAM orientation will form the **soft-iron variation**. Additionally, axes misalignment of the individual magnetometers may shift or rotate the axis of measurement. Non-magnetic effects may further locally distort the TAM measurement, e.g. rotating the TAM too fast while collecting measurements, roughly for angular speeds greater than $\approx 150^\circ/\text{s}$. However, careful calibration of the TAM can greatly reduce these errors.

To compute the yaw angle, ψ , one can use a TAM as

$$\vec{h}_B = \begin{bmatrix} h_{B,1} \\ h_{B,2} \\ h_{B,3} \end{bmatrix} \quad (19.9)$$

which can be projected into the horizontal plane through **tilt compensation (TC)** as

$$\vec{h}_{TC} = \begin{bmatrix} h_{TC,1} \\ h_{TC,2} \\ h_{TC,3} \end{bmatrix} = \begin{bmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \phi & -\sin \phi \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \vec{h}_B \quad (19.10)$$

which requires the current roll and pitch estimates to compute the projection. Then, the angle between the leveled measurements and magnetic north can be used to estimate the yaw angle, i.e.

$$\hat{\psi} = -\tan^{-1} \left(\frac{h_{TC,2}}{h_{TC,1}} \right) + \eta \quad (19.11)$$

where η is the magnetic declination between true north and magnetic north which must be obtained from a magnetic field map based on nominal position.

Attitude Determination from Rotational Rate

A **gyroscope** is a sensor that directly measures the rotational rate about an axis. Conceptually, it is a spinning wheel or disc in which the spin axis is free to assume any orientation by itself. When rotating, the orientation of this axis is unaffected by tilting or rotation of the mounting, according to the conservation of angular momentum. There are two broad design categories for gyroscopes: gyrostabilized and strapdown. **Gyrostabilized** measure the angles of gimbal bearings or fluid bearings. **Strapdown** measure either optical effects or vibrating structures effects to infer the rotational rate.

The Euler angles can be related to the angular velocity of the body-to-navigation frame $\vec{\omega}_{B,N \leftarrow B}$ which can be described as the difference between the inertial frames, i.e.

$$\vec{\omega}_{B,N \leftarrow B} = \vec{\omega}_{B,I \leftarrow B} - \vec{\omega}_{B,I \leftarrow N} \quad (19.12)$$

where $\vec{\omega}_{B,I \leftarrow B}$ can be measured using a three-axis gyroscope as

$$\vec{\omega}_{B,I \leftarrow B} = \begin{bmatrix} p_g \\ q_g \\ r_g \end{bmatrix} + \vec{b}_g \quad (19.13)$$

where \vec{b}_g is the gyroscope bias and $\vec{\omega}_{B,I \leftarrow N}$ can be computed as a function of the current latitude ℓ and longitude λ and their rates, i.e.

$$\vec{\omega}_{B,I \leftarrow N} = C_{B \rightarrow N} \begin{bmatrix} (\dot{\ell} + \omega_{I \leftarrow E}) \cos \lambda \\ -\dot{\lambda} \\ -(\dot{\ell} + \omega_{I \leftarrow E}) \sin \lambda \end{bmatrix} \quad (19.14)$$

where $\omega_{I \leftarrow E}$ is the Earth rotation rate, often taken as the WGS84 value 7.292115×10^{-5} rad/s. For automotive/consumer grade IMUs, one can typically neglect $\vec{\omega}_{B,I \leftarrow N}$ due to the much higher bias errors.

By defining the Euler angle vector as

$$\vec{\psi}_{N \leftarrow B} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (19.15)$$

Then, the Euler angle time derivatives can be related to the gyroscope measurements as

$$\dot{\vec{\psi}}_{N \leftarrow B} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos \theta} \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \theta \cos \phi & -\sin \phi \cos \theta \\ 0 & \sin \theta & \cos \phi \end{bmatrix} \begin{bmatrix} p_g \\ q_g \\ r_g \end{bmatrix} \quad (19.16)$$

which must be numerically integrated to determine the Euler angles. With proper initialization, this setup can be within several degrees of accuracy for tens of seconds without aiding even for low grade gyroscopes. However, this integration of bias alongside the measurements causes unbounded growth of the Euler angle estimates which requires aiding using additional attitude sensors, i.e. one requires sensor data fusion.

References

For more information, please refer to the following

- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “7 Digital Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 584-620

19.2 Gyroscopic Attitude Determination Systems

Attitude Determination from Rotational Rate

A **rate gyroscope** is a sensor that directly measures the rotational rate about an axis. Conceptually, it is a spinning wheel or disc in which the spin axis is free to assume any orientation by itself. When rotating, the orientation of this axis is unaffected by tilting or rotation of the mounting, according to the conservation of angular momentum. There are two broad design categories for gyroscopes: gyrostabilized and strapdown. **Gyrostabilized** measure the angles of gimbal bearings or fluid bearings. **Strapdown** measure either optical effects or vibrating structures effects to infer the rotational rate.

The Euler angles can be related to the angular velocity of the body-to-navigation frame $\vec{\omega}_{B,N \leftarrow B}$ which can be described as the difference between the inertial frames, i.e.

$$\vec{\omega}_{B,N \leftarrow B} = \vec{\omega}_{B,I \leftarrow B} - \vec{\omega}_{B,I \leftarrow N} \quad (19.17)$$

where $\vec{\omega}_{B,I \leftarrow B}$ can be measured using a three-axis gyroscope as

$$\vec{\omega}_{B,I \leftarrow B} = \begin{bmatrix} p_g \\ q_g \\ r_g \end{bmatrix} + \vec{b}_g \quad (19.18)$$

where \vec{b}_g is the gyroscope bias and $\vec{\omega}_{B,I \leftarrow N}$ can be computed as a function of the current latitude ℓ and longitude λ and their rates, i.e.

$$\vec{\omega}_{B,I \leftarrow N} = C_{B \rightarrow N} \begin{bmatrix} (\dot{\ell} + \omega_{I \leftarrow E}) \cos \lambda \\ -\dot{\lambda} \\ -(\dot{\ell} + \omega_{I \leftarrow E}) \sin \lambda \end{bmatrix} \quad (19.19)$$

where $\omega_{I \leftarrow E}$ is the Earth rotation rate, often taken as the WGS84 value 7.292115×10^{-5} rad/s. For automotive/consumer grade IMUs, one can typically neglect $\vec{\omega}_{B,I \leftarrow N}$ due to the much higher bias errors.

By defining the Euler angle vector as

$$\vec{\psi}_{N \leftarrow B} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (19.20)$$

Then, the Euler angle time derivatives can be related to the gyroscope measurements as

$$\dot{\vec{\psi}}_{N \leftarrow B} = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \frac{1}{\cos \theta} \begin{bmatrix} \cos \theta & \sin \theta \sin \phi & \sin \theta \cos \phi \\ 0 & \cos \theta \cos \phi & -\sin \phi \cos \theta \\ 0 & \sin \theta & \cos \phi \end{bmatrix} \begin{bmatrix} p_g \\ q_g \\ r_g \end{bmatrix} \quad (19.21)$$

which must be numerically integrated to determine the Euler angles. With proper initialization, this setup can be within several degrees of accuracy for tens of seconds without aiding even for low grade gyroscopes. However, this integration of bias alongside the measurements causes unbounded growth of the Euler angle estimates which requires aiding using additional attitude sensors, i.e. one requires sensor data fusion.

19.3 Single-Frame Attitude Determination Systems

Direct Attitude Determination

A **feature tracker** is a sensor that measures the features in the environment which can include the bearing, range, or position of passive or static objects and the strength or direction of a vector field, e.g. celestial bodies, edges, gravitational fields, or magnetic fields. Feature trackers can be a passive radio, infrared, or optical, or multi-spectral sensor or an active radar, sonar, or lidar. Using feature trackers provides the ability to compute a full attitude solution provided that two vectors can be measured in the body-fixed frame, \vec{u}_B and \vec{v}_B , and are known in the relative reference frame for attitude, e.g. the LVLH frame, \vec{u}_L and \vec{v}_L . From these vectors, one can form the following two pairs of orthogonal vector quantities to \vec{u}_B and \vec{u}_L as

$$\vec{r}_B = \frac{[\vec{u}_B] \times \vec{v}_B}{\|[\vec{u}_B] \times \vec{v}_B\|_2} \quad \vec{r}_L = \frac{[\vec{u}_L] \times \vec{v}_L}{\|[\vec{u}_L] \times \vec{v}_L\|_2} \quad (19.22)$$

$$\vec{s}_B = [\vec{u}_B] \times \vec{r}_B \quad \vec{s}_L = [\vec{u}_L] \times \vec{r}_L \quad (19.23)$$

Then, one can form the body matrix as

$$M_B = [\vec{u}_B \quad \vec{r}_B \quad \vec{s}_B] \quad (19.24)$$

and the reference matrix as

$$M_L = [\vec{u}_L \quad \vec{r}_L \quad \vec{s}_L] \quad (19.25)$$

which provides the relationship

$$M_B = C_{B \leftarrow L} M_L \quad (19.26)$$

which can be inverted to form the DCM estimate

$$\hat{C}_{B \leftarrow L} = M_B M_L^{-1} \quad (19.27)$$

which has covariance matrix

$$P_{\hat{C}} = \sigma_{\vec{u}}^2 I_{3 \times 3} + \frac{1}{\|\vec{u}_B \times \vec{v}_B\|_2^2} \left(\left(\sigma_{\vec{v}}^2 - \sigma_{\vec{u}}^2 \right) \vec{u}_B \vec{u}_B^T + \sigma_{\vec{u}}^2 (\vec{u}_B \cdot \vec{v}_B) \left(\vec{u}_B \vec{v}_B^T + \vec{v}_B \vec{u}_B^T \right) \right) \quad (19.28)$$

where $\sigma_{\vec{u}}^2$ is the variance of the \vec{u} measurement and $\sigma_{\vec{v}}^2$ is the variance of the \vec{v} measurement.

When more than two vectors quantities are present, then one can use an optimal parameter estimation algorithm to estimate the direction cosine matrix (DCM) or the quaternions. In this case, the weighted least-squares formulation of the attitude determination problem is known as **Wahba's problem** which states the following minimization for the DCM

$$\hat{C}_{B \leftarrow L} = \underset{C_{B \leftarrow L}}{\operatorname{argmin}} \frac{1}{2} \sum_{k=1}^N w_k \|\vec{b}_{B,k} - C_{B \leftarrow L} \vec{b}_{L,k}\|_2^2 \quad \text{for } N \geq 2 \quad (19.29)$$

where w_k are the individual weights of the N vector measurements, $\vec{b}_{B,k}$ is the vector measurement for the k^{th} feature in the body-fixed frame, and $\vec{b}_{L,k}$ is the reference vector for the k^{th} feature in the LVLH frame. This can also be rewritten in a more convenient form using the definition

$$B = \sum_{k=1}^N a_k \vec{b}_{B,k} \vec{b}_{L,k}^T \quad (19.30)$$

which provides

$$\hat{C}_{B \leftarrow L} = \underset{C_{B \leftarrow L}}{\operatorname{argmin}} \sum_{k=1}^N a_k - \operatorname{Tr} \left(C_{B \leftarrow L} B^T \right) \quad (19.31)$$

or

$$\hat{C}_{B \leftarrow L} = \underset{C_{B \leftarrow L}}{\operatorname{argmax}} \operatorname{Tr} \left(C_{B \leftarrow L} B^T \right) \quad (19.32)$$

The SVD solution method is to determine the singular value decomposition (SVD) of the non-square matrix B , i.e.

$$B = U \Sigma V^T \quad (19.33)$$

where U is an 3×3 orthogonal matrix, V is an 3×3 orthogonal matrix, and Σ is a diagonal 3×3 matrix with *non-negative* real numbers on the diagonal, i.e.

$$\Sigma = \begin{bmatrix} \sigma_1 & 0 & 0 \\ 0 & \sigma_2 & 0 \\ 0 & 0 & \sigma_3 \end{bmatrix} \quad (19.34)$$

The diagonal entries of Σ are known as the **singular values** of B and are ordered in size with $\sigma_1 \geq \sigma_2 \geq \sigma_3$. The singular values are also the square root of the eigenvalues of $B^T B$ or BB^T and are defined for all matrices, even non-square.

Then, the estimate of the DCM can be computed as

$$\hat{C}_{B \leftarrow L} = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \det(U) \det(V) \end{bmatrix} \quad (19.35)$$

with covariance

$$P_{\hat{C}} = U \begin{bmatrix} (\sigma_2 + \det(U) \det(V) \sigma_3)^{-1} & 0 & 0 \\ 0 & (\det(U) \det(V) \sigma_3 + \sigma_1)^{-1} & 0 \\ 0 & 0 & (\sigma_1 + \sigma_2)^{-1} \end{bmatrix} U^T \quad (19.36)$$

Other solution methods to Wahba's problem use quaternions methods to compute the optimal, e.g. the *q* and QUaternion ESTimator (QUEST) methods. The following two subsections discuss how one can use only one vector measurement from a single feature tracker to obtain a direct partial attitude determination.

Notably, for spacecraft, many different features can be tracked including the Earth, the sun, the moon, and the stars. A **star tracker** is a sensor which uses optical photocells or cameras to sense the magnitude of brightness and spectral type to identify and measure the relative bearing vectors to specific identifiable stars, known as **navigational stars**. Thus, if one has a **star map** defined for the bearing vectors to identifiable stars in a known reference frame, e.g. the ICRF, ECI, or ECEF frame, then one can use these bearing vectors in the body-fixed frame to solve for the attitude of the spacecraft in the body-fixed frame relative to the LVLH frame by first transforming the reference bearing vectors to the LVLH or navigation frame using the estimated position of the vehicle first which couples the positioning and attitude determination problems.

GNSS Phase-Based Attitude Estimation

GNSS phase-based attitude estimation is an extension of differential carrier phase positioning with the parameter to estimate as the baseline vector but with an additional constraint for a constant baseline, or nearly constant due to elastic motion of the vehicle. This very precise relative position estimate between a pair of antennas allows one to transform this relative position into angular measurements of the baseline vector. Thus, two baselines composed of three non-collinear antennas can completely define the attitude, e.g. the Euler angles.

For one baseline, recall that the double-differenced carrier phase range equation

$$\nabla_j \Delta_{RB} \Phi_i = \Delta_{RB} r_i + \lambda \nabla_j \Delta_{RB} N_i + \Delta_{RB} \epsilon_i \quad (19.37)$$

which can be shown to be related to the constant **baseline vector**, i.e.

$$\vec{x}_{RB} = \begin{bmatrix} x_{RB} \\ y_{RB} \\ z_{RB} \end{bmatrix} = \begin{bmatrix} x_R - x_B \\ y_R - y_B \\ z_R - z_B \end{bmatrix} \quad (19.38)$$

as

$$\nabla_j \Delta_{RB} \Phi_i = \frac{(\vec{x}_i - \vec{x})^T}{\|\vec{x} - \vec{x}_i\|_2} \vec{x}_{RB} + \lambda \nabla_j \Delta_{RB} N_i + \nabla_j \Delta_{RB} \epsilon_i \quad (19.39)$$

Thus, in phase-based attitude estimation, one has the parameter vector for $m + 1$ transmitters

$$\vec{\beta} = \begin{bmatrix} x_{RB} \\ y_{RB} \\ z_{RB} \\ \nabla_j \Delta_{RB} N_1 \\ \vdots \\ \nabla_j \Delta_{RB} N_m \end{bmatrix} \quad (19.40)$$

which has $m + 3$ unknowns with m DD measurements. Thus, one requires the use of multiple time steps in order to first resolve the integer ambiguities, $\nabla_j \Delta_{RB} N_i$. However, the known constraint of the baseline allows the fixed solution to resolve the integer ambiguities quicker from the float, e.g. **baseline-constrained LAMBDA method**. Typically, one uses a short-baseline to use the same LOS vectors for all three receivers in the NLS algorithm and reduce the number of candidate sets of integers.

However, once one has resolved the integer ambiguities, one can form the OLS solution for the baseline vector estimate as

$$\hat{\vec{\beta}} = (\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T (\nabla_j \Delta_{RB} \Phi_i - \lambda \nabla_j \Delta_{RB} N_i) \quad (19.41)$$

where

$$\mathbf{X} = \begin{bmatrix} \frac{(\vec{x}_1 - \vec{x})^T}{\|\vec{x} - \vec{x}_1\|_2} \\ \vdots \\ \frac{(\vec{x}_m - \vec{x})^T}{\|\vec{x} - \vec{x}_m\|_2} \end{bmatrix} \quad (19.42)$$

Furthermore, for two baselines with one rover denoted by the subscript $R1$ and the second rover denoted by the subscript $R2$, one can write

$$[\hat{\vec{x}}_{N,R1B} \quad \hat{\vec{x}}_{N,R2B} \quad \hat{\vec{x}}_{N,R1B} \times \hat{\vec{x}}_{N,R2B}] = C_{N \leftarrow B} [\vec{x}_{B,R1B} \quad \vec{x}_{B,R2B} \quad \vec{x}_{B,R1B} \times \vec{x}_{B,R2B}] \quad (19.43)$$

Thus, one can form the DCM estimate from

$$\hat{C}_{N \leftarrow B} = [\hat{\vec{x}}_{N,R1B} \quad \hat{\vec{x}}_{N,R2B} \quad \hat{\vec{x}}_{N,R1B} \times \hat{\vec{x}}_{N,R2B}] [\vec{x}_{B,R1B} \quad \vec{x}_{B,R2B} \quad \vec{x}_{B,R1B} \times \vec{x}_{B,R2B}]^{-1} \quad (19.44)$$

and the Euler angle estimates can be computed as

$$\begin{bmatrix} \hat{\phi} \\ \hat{\theta} \\ \hat{\psi} \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{c_{32}}{c_{33}} \\ -\sin^{-1} \frac{c_{31}}{c_{33}} \\ \tan^{-1} \frac{c_{21}}{c_{11}} \end{bmatrix} \quad (19.45)$$

where c_{ij} is the element in the i^{th} row and j^{th} column of $\hat{C}_{N \leftarrow B}$.

References

For more information, please refer to the following

- Stevens, B. L., Lewis, F. L., and Johnson, E. N., “7 Digital Control,” *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems*, 3rd ed., Vol. 1, Wiley-Blackwell, New York, 2015, pp. 584-620

19.4 Attitude and Heading Reference Systems

The most common type of attitude determination system which employs multi-sensor information fusion uses magnetometers, gyroscopes, and accelerometers to estimate the attitude and heading referenced to the Earth’s surface, i.e. the local navigation frame. This type of system is known as an **attitude and heading reference system (AHRS)**, also known as a **magnetic, angular rate, and gravity (MARG) system**. Furthermore, one can form an **Air Data AHRS (ADAHRS)** which uses the additional air data measurements to improve the models in an AHRS, typically by including a barometric altitude measurement for vertical channel stability and an air data triplet measurement for additional error modeling. For example, if one is using an ADAHRS which provides an airspeed measurement, one can estimate the total acceleration with the rotational acceleration, which is typically much larger than the translational acceleration, as

$$\vec{a}_B = \dot{\vec{v}}_B + [\vec{\omega}_{I \leftarrow B, B}] \times \vec{v}_B \quad (19.46)$$

$$\vec{a}_B \approx [\vec{\omega}_{I \leftarrow B, B}] \times \vec{v}_B \quad (19.47)$$

$$\vec{a}_B \approx [\vec{\omega}_{I \leftarrow B, B}] \times \vec{v}_B \quad (19.48)$$

$$\vec{a}_B \approx \begin{bmatrix} 0 \\ v_\infty r_g \\ -v_\infty q_g \end{bmatrix} \quad (19.49)$$

where v_∞ is the airspeed and q_g and r_g are the gyroscope measurements of the angular velocity about the y body axis and z body axis, respectively. Notably, this correlates the gyroscope angular velocity measurements used in the nominal-state estimation with the airspeed-based error-state estimation. The primary challenges with an AHRS are non-gravitational vehicle accelerations which are indistinguishable from vehicle tilt errors and magnetic field disturbances due to onboard electronics. Both error sources introduce unknown time-correlated errors into the aiding measurements, violating the white noise assumption in Bayesian filtering. However, with proper error modeling and tuning, one can derive an (AD)AHRS using low grade IMUs, magnetometers, and optional air data sensors.

ES-EKF for AHRS

As the 3-DOF attitude dynamics in navigation are highly nonlinear integrations of Euler angles, DCMs, or quaternions, one often uses **error-state filtering** as the multi-sensor data fusion algorithm, which uses a time update step for the gyroscope nominal-state and the gyroscope error-state and measurement update steps for the error-state using the accelerometer and magnetometer measurements. This but includes an additional state update step for correcting the nominal-state estimate with the error-state estimate. In navigation, the *de*

facto standard algorithm for data fusion is the error-state EKF although the UKF and the PF have been used in some cases. This section will describe the error-state Extended Kalman filter (ES-EKF) implementation in an AHRS.

For an AHRS, the estimated nominal-state contains the three estimated nominal Euler angles $\hat{\psi}_{N \leftarrow B}$ and the three estimated nominal gyroscope biases \hat{b}_g , i.e.

$$\hat{x} = \begin{bmatrix} \hat{\psi}_{N \leftarrow B} \\ \hat{b}_g \end{bmatrix} \quad (19.50)$$

where the measured gyroscope angular rates, $\vec{\omega}_{B,N \leftarrow B}$ are corrected for the nominal-biases, \hat{b}_g , and integrated to obtain the Euler angles, $\hat{\psi}_{N \leftarrow B}$. The error-state vector contains the three Euler angle errors $\delta \vec{\psi}_{N \leftarrow B}$ and three gyroscope bias errors $\delta \vec{b}_g$, i.e.

$$\delta \vec{x} = \begin{bmatrix} \delta \vec{\psi}_{N \leftarrow B} \\ \delta \vec{b}_g \end{bmatrix} \quad (19.51)$$

For the ES-EKF state update step, the gyroscope nominal-state bias estimates are simply updated by

$$\hat{b}_g = \hat{b}_g - \delta \hat{b}_g \quad (19.52)$$

For the Euler angle attitude update, one cannot must use the corresponding DCM error model. Using a first-order approximation for the DCM error, one has

$$C_{N \leftarrow B} = \left(I_{3 \times 3} + \left[\delta \vec{\psi}_{N \leftarrow B} \right]^\times \right) C_{N \leftarrow B} \quad (19.53)$$

Then, the updated Euler angles can be extracted from $C_{N \leftarrow B}$ by

$$\hat{\psi}_{N \leftarrow B} = \begin{bmatrix} \hat{\phi} \\ \hat{\theta} \\ \hat{\psi} \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{c_{32}}{c_{33}} \\ -\sin^{-1} \frac{c_{31}}{c_{33}} \\ \tan^{-1} \frac{c_{21}}{c_{11}} \end{bmatrix} \quad (19.54)$$

where c_{ij} is the element in the i^{th} row and j^{th} column of $C_{N \leftarrow B}$.

For the ES-EKF time update step, one can form linearized matrices based on the linearized Euler angle error equation, i.e.

$$\dot{\hat{\psi}}_{N \leftarrow B} \approx - [\vec{\omega}_{N,I \leftarrow N}]^\times \delta \vec{\psi}_{N \leftarrow B} + \delta \vec{\omega}_{N,I \leftarrow N} - C_{N \leftarrow B} \delta \vec{\omega}_{B,I \leftarrow B} \quad (19.55)$$

where one typically drops the two terms with the transport rate, $\delta \vec{\omega}_{N,I \leftarrow N}$, for automotive/consumer grade gyroscopes. The gyroscope errors are typically modeled with a first-order Markov process model as

$$\begin{aligned} \delta \vec{\omega}_{B,I \leftarrow B} &= \vec{b}_g + \vec{w}_g \\ \dot{\vec{b}}_g &= -\frac{1}{\tau_g} I_{3 \times 3} \vec{b}_g + \vec{\mu}_g \end{aligned} \quad (19.56)$$

which can be stacked as process noise vector

$$\vec{w} = \begin{bmatrix} \vec{w}_g \\ \vec{\mu}_g \end{bmatrix} \quad (19.57)$$

with power spectral density

$$S = \begin{bmatrix} \sigma_{b_g}^2 & 0 \\ 0 & \frac{2\sigma_{b_g}^2}{\tau_g} \end{bmatrix} \quad (19.58)$$

Taken together, the linearized error-state dynamics equation is

$$\delta \dot{\vec{x}} = A \delta \vec{x} + L \vec{w} \quad (19.59)$$

$$\begin{bmatrix} \delta \dot{\vec{\psi}}_{N \leftarrow B} \\ \dot{\vec{b}}_g \end{bmatrix} = A \begin{bmatrix} \delta \vec{\psi}_{N \leftarrow B} \\ \vec{b}_g \end{bmatrix} + L \begin{bmatrix} \vec{w}_g \\ \vec{\mu}_g \end{bmatrix} \quad (19.60)$$

where

$$A = \begin{bmatrix} 0 & -C_{N \leftarrow B} \\ 0 & -\frac{1}{\tau_g} I_{3 \times 3} \end{bmatrix} \quad (19.61)$$

and

$$L = \begin{bmatrix} -C_{N \leftarrow B} & 0 \\ 0 & I_{3 \times 3} \end{bmatrix} \quad (19.62)$$

For the ES-EKF measurement update, the error-state covariance requires the linearization of $C_{N \leftarrow B}$, to form for the error in the accelerometer-based gravity model

$$\delta \vec{g}_N = \vec{g}_N - \left(I_{3 \times 3} - [\delta \vec{\psi}_{N \leftarrow B}]^\times \right) C_{N \leftarrow B} (\vec{g}_B - \vec{a}_{B,I \leftarrow B} - \delta \vec{f}_B - \vec{v}_f) \quad (19.63)$$

or

$$\delta \vec{g}_N = [\delta \vec{\psi}_{N \leftarrow B}]^\times \vec{g}_N + [I_{3 \times 3} - \delta \vec{\psi}_{N \leftarrow B}]^\times C_{N \leftarrow B} (\vec{a}_{B,I \leftarrow B} + \delta \vec{f}_B + \vec{v}_f) \quad (19.64)$$

and dropping multiplicative error terms, using the fact that

$$[\vec{a}]^\times \vec{b} = -[\vec{b}]^\times \vec{a} \quad (19.65)$$

and combining the errors into a single term $\vec{v}_a = \vec{a}_{B,I \leftarrow B} + \delta \vec{f}_B + \vec{v}_f$ with covariance R_a , one has the accelerometer measurement equation

$$\delta \vec{g}_N = -[\vec{g}_N]^\times \delta \vec{\psi}_{N \leftarrow B} - C_{N \leftarrow B} \vec{v}_a \quad (19.66)$$

or

$$\delta \vec{g}_N = \begin{bmatrix} 0 & \|g_N\| & 0 \\ -\|g_N\| & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \delta \vec{\psi}_{N \leftarrow B} + C_{N \leftarrow B} \vec{v}_a \quad (19.67)$$

which demonstrates, as expected, only roll and pitch are observable by the accelerometer.

For the measurement update of the roll error estimate, $\delta\hat{\psi}$, one has a linearized error measurement equation

$$\delta\vec{h}_N = \vec{h}_N - \left(I_{3 \times 3} - [\delta\vec{\psi}_{N \leftarrow B}]^\times \right) C_{N \leftarrow B} (\vec{h}_B + \vec{v}_h) \quad (19.68)$$

which

$$\delta\vec{h}_N = -[\vec{h}_N]^\times \delta\vec{\psi}_{N \leftarrow B} + C_{N \leftarrow B} \vec{v}_h \quad (19.69)$$

$$\delta\vec{h}_N = \begin{bmatrix} 0 & h_{N,3} & -h_{N,2} \\ -h_{N,3} & 0 & h_{N,1} \\ h_{N,2} & -h_{N,1} & 0 \end{bmatrix} \delta\vec{\psi}_{N \leftarrow B} + C_{N \leftarrow B} \vec{v}_h \quad (19.70)$$

where \vec{v}_h has covariance R_h and accounts for the combined magnetometer measurement noise, magnetic field disturbances, and residual errors in the sensor calibration.

The AHRS ES-EKF measurement model is for the innovation $\delta\vec{y}_k$, i.e.

$$\delta\vec{y}_k = \begin{bmatrix} \vec{g}_N \\ \vec{h}_N \end{bmatrix} - \begin{bmatrix} C_{N \leftarrow B,k} \hat{\vec{g}}_B \\ C_{N \leftarrow B,k} \hat{\vec{h}}_B \end{bmatrix} \quad (19.71)$$

The error-state linearized measurement matrix is

$$H_k = \begin{bmatrix} 0 & \|g_N\| & 0 \\ -\|g_N\| & 0 & 0 \\ 0 & 0 & 0 \\ 0 & h_{N,3} & -h_{N,2} \\ -h_{N,3} & 0 & h_{N,1} \\ h_{N,2} & -h_{N,1} & 0 \end{bmatrix} \quad (19.72)$$

and the error-state linearized measurement noise matrix is

$$M_k = \begin{bmatrix} C_{N \leftarrow B,k} \\ C_{N \leftarrow B,k} \end{bmatrix} \quad (19.73)$$

and the stacked measurement noise vector is

$$\vec{v} = \begin{bmatrix} \vec{v}_a \\ \vec{v}_h \end{bmatrix} \quad (19.74)$$

with stacked covariance matrix

$$\mathbf{R} = \begin{bmatrix} R_a & 0 \\ 0 & R_h \end{bmatrix} \quad (19.75)$$

where $C_{N \leftarrow B,k}$ depends on some low-rate sampling of the Euler angle estimates from the integrated gyroscope.

Positioning Systems

20.1 Introduction to Positioning Systems

Vehicle perception systems that consider the problem of estimating the position of a vehicle relative to a predefined reference frame are said to be performing **positioning**. One potential reference frame could simply be chosen as the origin of some movement and the vehicle's position is then computed relative to this beginning by estimating the change in movement. Positioning using only the integration of the vehicle's movement or velocity is known as **dead reckoning (DR)** which is fully presented in the following chapter on navigation. **Navigation** is the estimation of a vehicle's position, velocity, *and* attitude, which together are also known as the vehicle's **navigation state**. It should be noted that the primary drawback of DR results from the compounding of velocity estimation errors through integration. Thus, the positioning error in DR increases unboundedly over time. Lastly, it should be noted that the vehicle's position *and* attitude are also known as the vehicle's **pose** whose combined estimation may also be considered positioning.

Importantly, positioning and navigation typically require one to be able to also sense and estimate any potential changes in the sources of measurements as well as any measurement signal processing computations. Thus, any positioning or navigation system will also require accurate time measurements provided by a clock as part of a timing system. Thus, due to their interrelationship, these three quantities are often considered as the single engineering discipline of **positioning, navigation, and timing (PNT)** in vehicle information systems.

Related to reference frames in positioning is the potential use of a **map** which defines the positions of collective features within a reference frame. In this case, one is said to be performing **localization** if one describes the **location** of vehicle relative to these collective objects. This type of localization may be qualitative or quantitative, but a position is always defined as the *coordinates* within a defined reference frame.

Positioning Measurements and Sensors

The four fundamental positioning measurement types are **bearing**, i.e. the direction from the measurement source to the vehicle, **range**, i.e. the distance from the measurement source to the vehicle, **differential range**, i.e. the relative distance from two measurement sources to the vehicle, and **range rate**, i.e. the instantaneous change in the distance from the measurement source to the vehicle. These first three measurement types allow one to form corresponding **lines-of-position (LOP)** relating the measurement source(s) and the vehicle. When using only bearing LOPs, one can perform positioning using **multiangulation** techniques and when using only range or differential range LOPs, one can perform positioning using **multilateration** techniques. Using bearings and ranges LOPs, one can perform positioning using **multiangulateration** techniques. Other types of measurements used in positioning are differential ranges and range rates.

Measurement sources can generally be considered as either intentionally-designed **beacons** or identifiable environmental **features**. In beacon-based positioning, a vehicle uses sensors to detect, identify, and process the *known* signals of the beacons which have known characteristics convenient for positioning algorithms. Common beacons include radio beacons and visible light beacons, e.g. lighthouses. In feature-based positioning, a vehicle uses sensors to detect, identify, and process the signals from environmental features in order to perform positioning within one's environment. Types of environmental features that are commonly used for positioning are topographical structures, magnetic fields, celestial bodies, atmospheric and ocean pressure, and exploitable radio signals also known as **signals of opportunity (SOP)**. Pre-modern positioning primarily used known topographical structures using eyesight, the Earth's magnetic field using compasses, and the observable movement of celestial bodies using eyesight.

Of note, until the advent of electronic inertial sensors, which can be used to perform accurate dead reckoning, navigators used the geographic position of celestial bodies to perform open ocean positioning. The **geographic position (GP)** of a celestial body is the point on the Earth's surface from which the body is currently directly overhead. However the computation of the GP requires celestial trigonometry and is also known as **sight reduction**. Thus by careful observation and calculation of the movement of the celestial body, celestial maps were created and used to be able to determine a celestial body's current GP. However, sufficient determination for reliable navigation was only made possible by the invention of accurate clocks with sufficient stability during long sea voyages. Furthermore as most celestial body GP maps were tabulated to GMT, it was standard to set one's clock to **Greenwich Mean Time (GMT)** where the prime meridian, i.e. 0° longitude, passes through Greenwich, England. In modern times, GMT has become the time standard UTC. Using this GP information, the **sextant** allowed one to measure the angle between the horizon in the direction of the GP and its corresponding celestial object in the sky. This angle can be shown to be directly proportional to the range of the navigator to the GP and this angle-to-distance conversion forms the origin of the distance unit of the nautical mile as equal to one minute of latitude.

With modern electronic sensors, positioning primarily uses energy wave signals, e.g. radio, light, acoustic. One uses **antennas** to sense radio signals, **photosensors**, also known as **photodetectors**, to sense ultraviolet (UV), visible, or near infrared (NIR) light signals, and **sonar** to sense transmitted and received acoustic signals. These sensors convert the sensed wave signal into an electrical signal for use in a signal processing algorithm before use in a positioning algorithm. This signal processing can use several characteristics of these received signals to derive an LOP. One may use the observed **angle of arrival (AOA)** of the received signal to derive a bearing measurement. One may also use the observed **time of flight (TOF)** of a transmitted *and* received signal to derive a range measurement, e.g. **radar**.

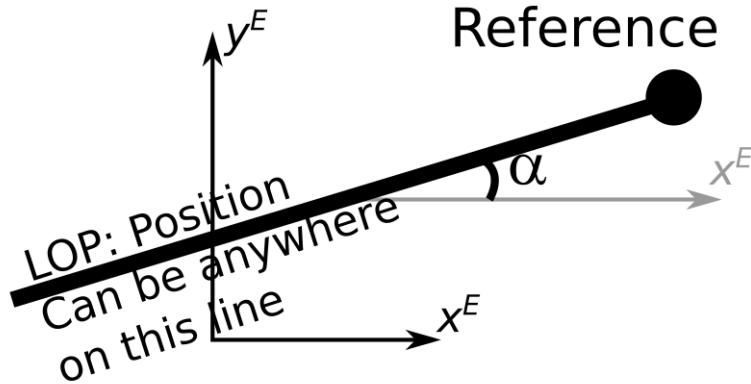
However, if one is simply receiving a *known* univariate signal, one may use the observed **time of arrival**

(TOA) of the received signal to derive a **pseudorange** measurement, i.e. “range” based on the timing of transmitted signals introduces additional timing errors in the measurement that must be estimated. One may also use the observed **time difference of arrival (TDOA)** of two received wave signals to derive a **differential pseudorange**. Another alternative is to use the observed **phase difference** for additional timing information for ranging. For range rate measurements using *known* univariate signals, one typically use Doppler effect of energy waves. The **Doppler effect** describes the increase/decrease in the frequency of an energy wave as the source and observer move toward/away from each other.

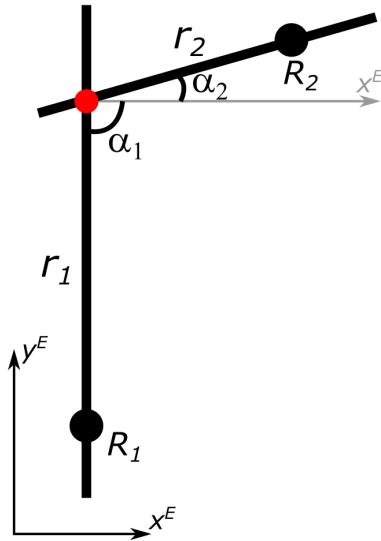
For the signal processing of two- or three-dimensional signals, one can produce **images** or **point clouds**, when often uses additional signal processing algorithms known as **image processing** or **computer vision** algorithms to derive bearing, range, and/or range rate measurements. In addition to these energy wave signals, one may also incorporate other sensors to aid in the positioning. One can use pressure gauges to sense air or water pressure, magnetometers to sense the Earth’s magnetic field, and/or radiometers, e.g. WiFi signals, which allow one to localize one’s position based on maps of the pressure, magnetic field and/or passive radio signals. One may also use inertial sensors and/or mechanical linkages which allow one to form proprioceptive sensors for estimating changes in position motion, i.e. movement.

Bearing Triangulation

A bearing LOP from a measurement source in two-dimensions, α , can be visualized as a straight line as shown in the following figure.



which results in identifying where the two bearing LOPs intersect as shown in the following figure with two sources of bearing measurements α_1 and α_2

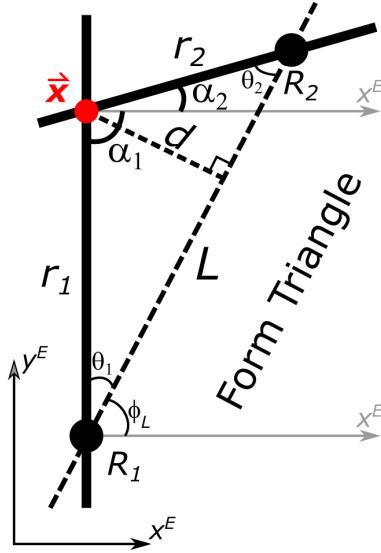


When using two bearing measurements, one can perform positioning using **triangulation** comes from the fact that the angles of the triangle are known as well as one of the sides known as the **baseline** between reference sources R_1 and R_2 . For more than two sources of bearing measurements, one can perform positioning using **multiangulation** which will increase the accuracy of the position estimate due to any uncertainty in the measurements, i.e. noise. Note that for long distances on Earth, multiangulation may require spherical trigonometry for calculating the LOPs.

To solve the two-dimensional triangulation problem geometrically, assume that one has a two-dimensional unknown position represented as a vector

$$\vec{x} = [x \ y]^T \quad (20.1)$$

and that one knows the reference positions of the source of the bearing angles, \vec{x}_{R1} and \vec{x}_{R2} . Furthermore, assume that one has knowledge of the absolute reference frame, the direction of x^E from which one has measured α_1 and α_2 . Then, one can form the following triangle.



Defining, the relative angle of baseline to x^E as

$$\phi_L = \tan^{-1} \frac{y_{R2} - y_{R1}}{x_{R2} - x_{R1}} \quad (20.2)$$

by trigonometry, one has as base angles of the triangle

$$\theta_1 = 180^\circ + \alpha_1 - \phi_L \quad (20.3)$$

as $\alpha_1 < 0^\circ$

$$\theta_2 = \phi_L - \alpha_2 \quad (20.4)$$

and a baseline length as

$$L = \|\vec{x}_{R1} - \vec{x}_{R2}\|_2 \quad (20.5)$$

one can write that

$$L = \frac{d}{\tan \theta_1} + \frac{d}{\tan \theta_2} \quad (20.6)$$

where d is the only unknown. This equation can be rearranged as

$$L = d \frac{\cos \theta_1}{\sin \theta_1} + d \frac{\cos \theta_2}{\sin \theta_2} \quad (20.7)$$

$$L = d \frac{\sin \theta_1 \cos \theta_2 + \sin \theta_2 \cos \theta_1}{\sin \theta_1 \sin \theta_2} \quad (20.8)$$

$$L = d \frac{\sin(\theta_1 + \theta_2)}{\sin \theta_1 \sin \theta_2} \quad (20.9)$$

$$d = L \frac{\sin \theta_1 \sin \theta_2}{\sin(\theta_1 + \theta_2)} \quad (20.10)$$

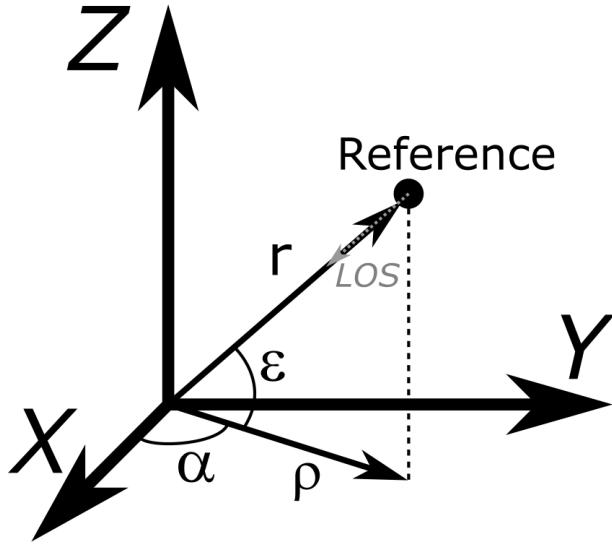
Now that one has d , one can find the position coordinates as

$$\vec{x} = \vec{x}_{R_1} + \begin{bmatrix} r_1 \sin \alpha_1 \\ r_1 \cos \alpha_1 \end{bmatrix} \quad (20.11)$$

$$\vec{x} = \vec{x}_{R_1} + \begin{bmatrix} \frac{d \sin \alpha_1}{\sin \theta_1} \\ \frac{d \cos \alpha_1}{\sin \theta_1} \end{bmatrix} \quad (20.12)$$

though one could also reference from \vec{x}_{R_2} .

For three-dimensional positioning, the bearing LOPs can be completely defined by two angles known as **azimuth**, $\alpha \in [0, 2\pi]$, and **elevation**, $\epsilon \in [0, \pi]$, as shown in the following figure for one measurement source,



Here, ρ is called the **ground range** and r is called the **slant range**.

To solve the three-dimensional multiangulation problem for an unknown position $\vec{x} = [x \ y \ z]^T$ and a reference located at $\vec{x}_R = [x_R \ y_R \ z_R]^T$, one has can form the **line-of-sight (LOS)** unit vector as

$$\frac{\vec{x} - \vec{x}_R}{\|\vec{x} - \vec{x}_R\|_2} = \begin{bmatrix} \cos \alpha \cos \epsilon \\ \sin \alpha \cos \epsilon \\ \sin \epsilon \end{bmatrix} \quad (20.13)$$

or using trigonometry, one has

$$\begin{bmatrix} \tan \alpha \\ \tan \epsilon \end{bmatrix} = \begin{bmatrix} \frac{x - x_R}{y - y_R} \\ \frac{z - z_R}{\sqrt{(x - x_R)^2 + (y - y_R)^2}} \end{bmatrix} \quad (20.14)$$

$$\begin{bmatrix} \alpha \\ \epsilon \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{x - x_R}{y - y_R} \\ \tan^{-1} \frac{z - z_R}{\sqrt{(x - x_R)^2 + (y - y_R)^2}} \end{bmatrix} \quad (20.15)$$

For this problem there are three unknown coordinates and two measurements for each reference. Thus, for two references, one has an over-constrained problem. As a solution, one could ignore one of the four measurements and solve this geometrically similarly to the two-dimensional case. However, as one typically must consider measurement noise, one can use a nonlinear estimation technique, e.g. nonlinear least-squares, to perform the positioning with the model chosen with a parameter vector as

$$\vec{\beta} = \vec{x} \quad (20.16)$$

a measurement vector as

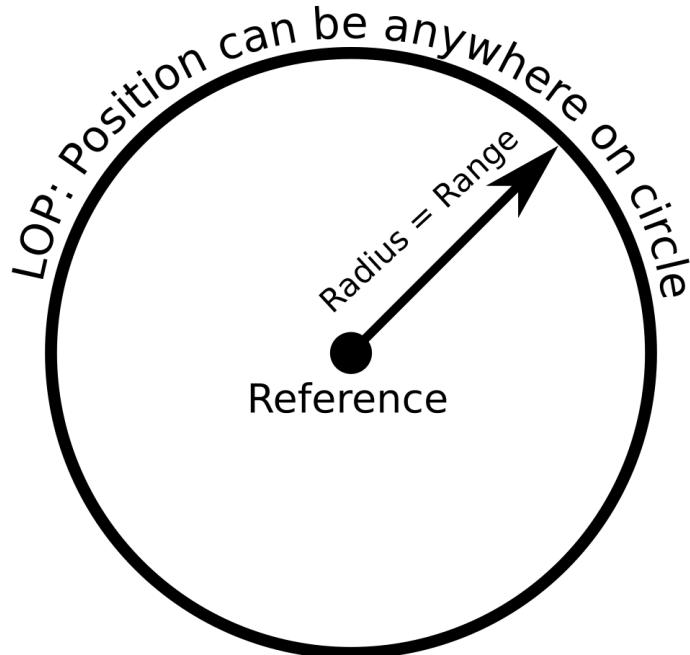
$$\vec{y} = \begin{bmatrix} \alpha_1 \\ \epsilon_1 \\ \alpha_2 \\ \epsilon_2 \end{bmatrix} \quad (20.17)$$

and a nonlinear measurement model as

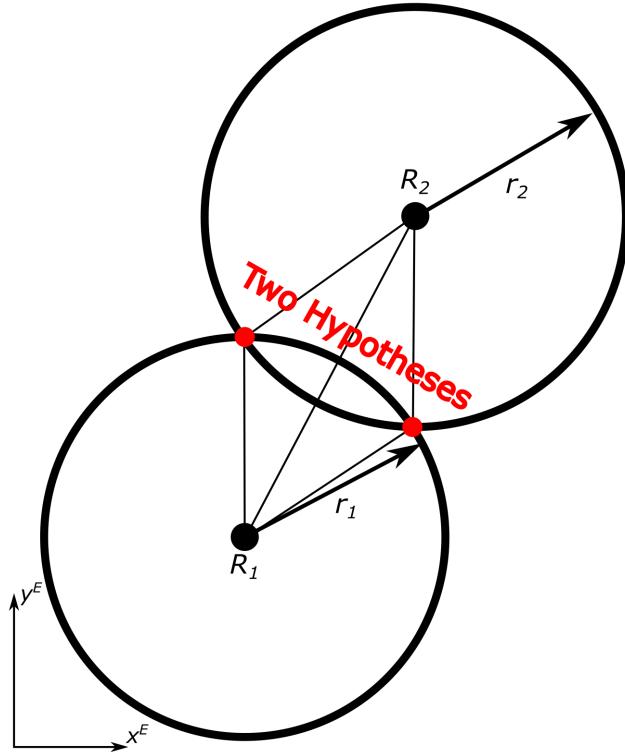
$$\vec{f}([\vec{x}_{R1} \ \vec{x}_{R2}]^T, \vec{x}) = \begin{bmatrix} \tan^{-1} \frac{x - x_{R1}}{y - y_{R1}} \\ \tan^{-1} \frac{z - z_{R1}}{\sqrt{(x - x_{R1})^2 + (y - y_{R1})^2}} \\ \tan^{-1} \frac{x - x_{R2}}{y - y_{R2}} \\ \tan^{-1} \frac{z - z_{R2}}{\sqrt{(x - x_{R2})^2 + (y - y_{R2})^2}} \end{bmatrix} \quad (20.18)$$

Range Trilateration

A range LOP from a measurement source in two-dimensional, r , can be visualized as a circle as shown in the following figure

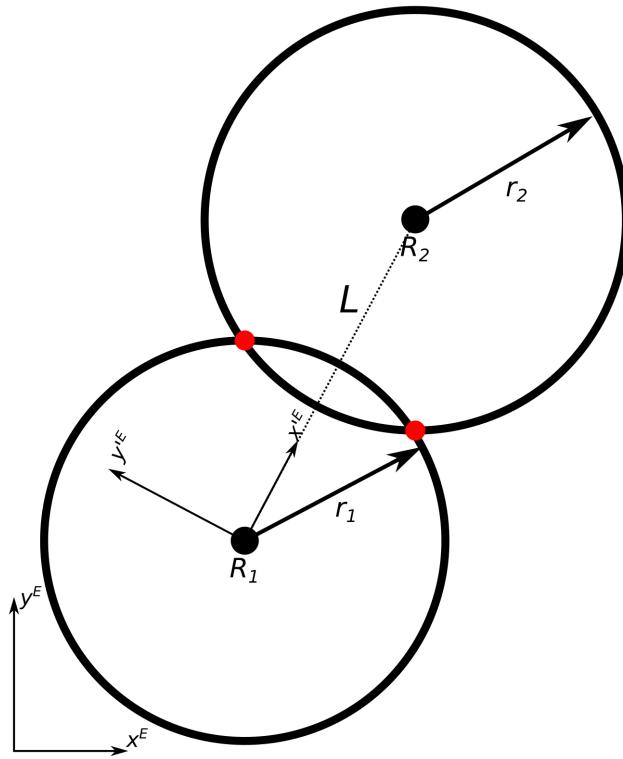


These LOPs provide an estimate of position using the intersection of circles shown here for two sources as



and with an initial guess, one can select the more likely hypothesis of the two possibilities. For each hypothesis, one knows the lengths of the sides of the associated triangle, thus this positioning is known as **trilateration**.

To solve the two-dimensional multilateration problem geometrically, one can form a “new” $x - y$ frame with the origin at R_1 and the x' -axis pointing at R_2 , i.e.



Next, by noting the baseline length is now

$$L = \|\vec{x}_{R1} - \vec{x}_{R2}\|_2 \quad (20.19)$$

one can write the range measurements in terms of the unknown coordinates as

$$r_1^2 = x'^2 + y'^2 \quad (20.20)$$

$$r_2^2 = (L - x')^2 + y'^2 \quad (20.21)$$

which can be solved simultaneously for the unknown position coordinates in the new frame as

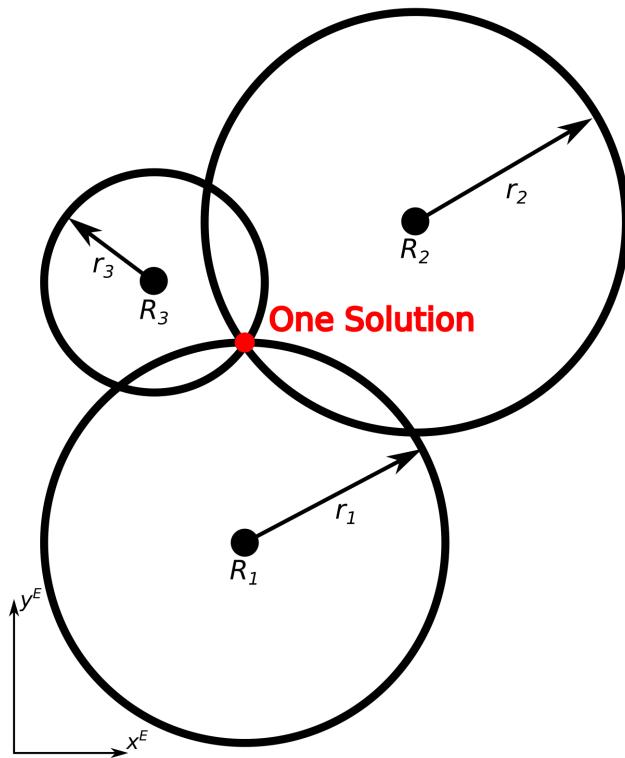
$$x' = \frac{r_1^2 - r_2^2 + L^2}{2L} \quad (20.22)$$

and

$$y' = \pm\sqrt{r_1^2 - x'^2} \quad (20.23)$$

and by transforming back to the original frame, \$x\$ and \$y\$ can be obtained.

With three or more references, one can obtain improved accuracy due to measurement noise which generalizes trilateration to multilateration positioning which is shown in the following figure



Lastly, it is important to note that in three-dimensions, range LOPs become spheres for which an intersection point requires at least three references for two position hypotheses from which one must infer which is correct.

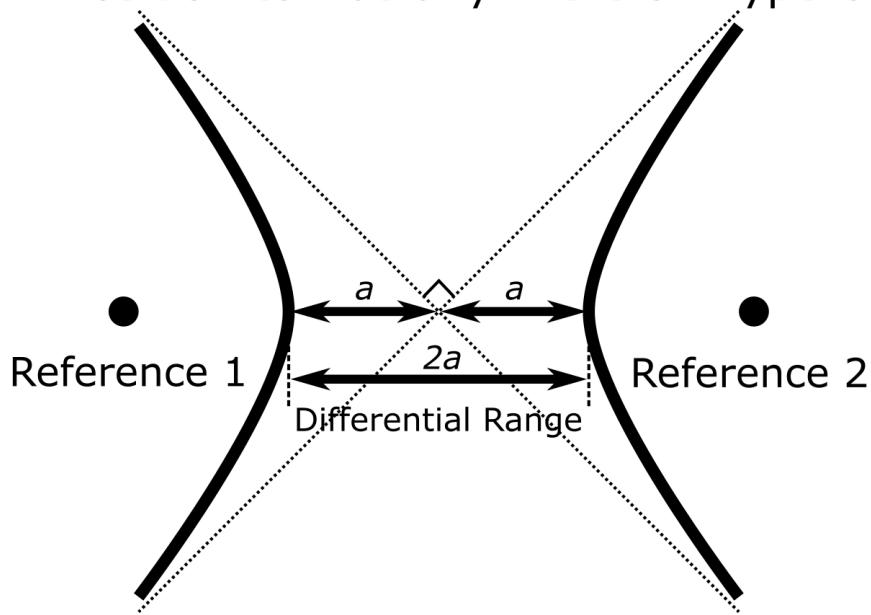
Differential Range Multilateration

A differential range LOP from two measurement source(s) in two-dimensional, $2a$, as

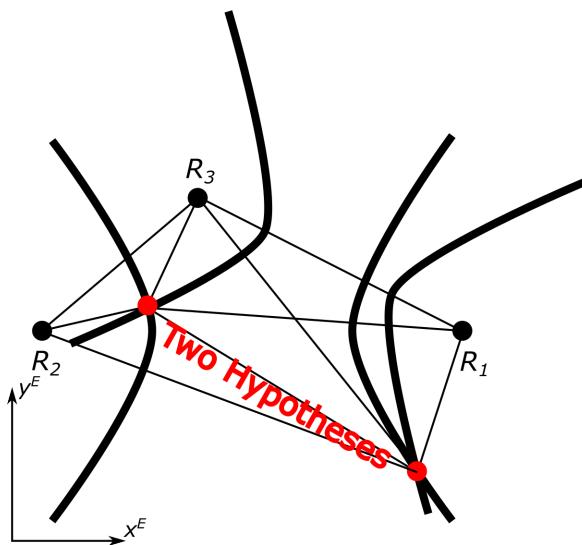
$$\|r_1\| - \|r_2\| = 2a \quad (20.24)$$

which can be visualized as a rectangular hyperbola as shown in the following figure

LOP: Position can be anywhere on hyperbola



These LOPs provide an estimate of position using the intersection of the rectangular hyperbolas shown here for three sources as



and with an initial guess, one can select the more likely hypothesis of the two possibilities. For each hypothesis, one knows the length of one side or the combined length of two sides of the two associated triangles in the diagram, thus this positioning is also known as **multilateration**, albeit, the trigonometric

relationships are more difficult than true-range trilateration and are left for the reader.

20.2 Pseudorange Multilateration

Un-Differenced Pseudorange Multilateration

In true range multilateration, one directly measures the Euclidean distance, i.e. the vector L_2 -norm, between the i^{th} transmitter and the signal receiver, i.e.

$$r_i = \|\vec{x} - \vec{x}_i\|_2 \quad (20.25)$$

which in three-dimensional space, i.e. $\vec{x} = [x \ y \ z]^T$ and $\vec{x}_i = [x_i \ y_i \ z_i]^T$, is written by component as

$$r_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} \quad (20.26)$$

In pseudorange multilateration, one measures the change in time of transmission (TOT) of the i^{th} signal, TOT_i , to the time of arrival (TOA), TOA_i . Thus, by multiplying difference between TOA and TOT by the assumed constant signal speed, c , one can form the **pseudorange**, ρ_i , for the i^{th} transmitter as

$$\rho_i = c(TOA_i - TOT_i) \quad (20.27)$$

However, this rough approximation to the true range is corrupted by errors, both known and unknown. Thus, one typically relates the pseudorange to the true range by the (**un-differenced**) **pseudorange equation** defined as

$$\rho_i = r_i + c(\delta t - \delta t_i) + b_{p,i} + b_{c,i} + \epsilon \quad (20.28)$$

where δt accounts for any clock error in TOA_i , δt_i accounts for any clock error in TOT_i , $b_{p,i}$ accounts for any distance error bias due to the assumed transmitter position, $b_{c,i}$ accounts for any distance error bias due to the assumed signal speed. Lastly, ϵ accounts for any additional unknown random errors, e.g. electrical noise. In pseudorange multilateration applications, one must typically refine the initially calculated ρ_i based on *transmitter-dependent* error models, δt_i , $b_{p,i}$, and $b_{c,i}$. Furthermore, δt is a receiver clock error that changes with time and must be estimated along with the receiver position coordinates in pseudorange multilateration. Removing the modeled error biases, one has the **simplified pseudorange equation** defined as

$$\rho_i = r_i - c\delta t + \epsilon \quad (20.29)$$

Substituting for r_i , one has

$$\rho_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} - c\delta t + \epsilon \quad (20.30)$$

where x , y , z , and δt are the four unknown parameters. Thus, in pseudorange multilateration, one has for the parameter vector

$$\vec{\beta} = \begin{bmatrix} x \\ y \\ z \\ c\delta t \end{bmatrix} \quad (20.31)$$

which implies that pseudorange multilateration requires four or more transmitters to be able to obtain a position estimate as opposed to three or more for true range multilateration. The simplified pseudorange equation allows one to form a nonlinear observation equation with m transmitters as

$$\vec{y} = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_m \end{bmatrix} = \mathbf{f}(\vec{x}, \hat{\beta}) = \begin{bmatrix} \sqrt{(\hat{x} - x_1)^2 + (\hat{y} - y_1)^2 + (\hat{z} - z_1)^2} + c\hat{\delta}t \\ \vdots \\ \sqrt{(\hat{x} - x_m)^2 + (\hat{y} - y_m)^2 + (\hat{z} - z_m)^2} + c\hat{\delta}t \end{bmatrix} \quad (20.32)$$

To solve this problem using NLS with GNA, assume one has an initial estimate, $\hat{\beta}_0$. Then, one can iteratively improve $\hat{\beta}$ by $\vec{\Delta}_k$ for $k = 0, 1, \dots$, i.e.

$$\hat{\beta}_{k+1} = \hat{\beta}_k + \vec{\Delta}_k \quad (20.33)$$

where

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\vec{y} - \mathbf{f}(\vec{x}, \hat{\beta}_k)) \quad (20.34)$$

where the i^{th} row of $m \times 4 \mathbf{J}$ can be written as

$$\mathbf{J}_i = \begin{bmatrix} \frac{1}{2} [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2]^{-1/2} 2(x - x_i) \\ \frac{1}{2} [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2]^{-1/2} 2(y - y_i) \\ \frac{1}{2} [(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2]^{-1/2} 2(z - z_i) \\ 1 \end{bmatrix}_{\vec{x}=\hat{\vec{x}}_k}^T \quad (20.35)$$

or simplifying, one has

$$\begin{aligned} \mathbf{J}_i &= \begin{bmatrix} \frac{x-x_i}{r_i} & \frac{y-y_i}{r_i} & \frac{z-z_i}{r_i} & 1 \end{bmatrix}_{\vec{x}=\hat{\vec{x}}_k} \\ &= \begin{bmatrix} \frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} & 1 \end{bmatrix}_{\vec{x}=\hat{\vec{x}}_k} \end{aligned} \quad (20.36)$$

where $\frac{\vec{x} - \vec{x}_i}{\|\vec{x} - \vec{x}_i\|_2}$ is notably the **line-of-sight (LOS)** unit vector from the transmitter to the receiver.

Note that if one assumes $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, where the covariance is approximately given by

$$\text{Cov}(\hat{\beta}) = \sigma^2 (\mathbf{J}^T \mathbf{J})^{-1} = \sigma^2 \begin{bmatrix} \sigma_x^2 & \sigma_{xy} & \sigma_{xz} & c\sigma_{xt} \\ \sigma_{xy} & \sigma_y^2 & \sigma_{yz} & c\sigma_{yt} \\ \sigma_{xz} & \sigma_{yz} & \sigma_z^2 & c\sigma_{zt} \\ c\sigma_{xt} & c\sigma_{yt} & c\sigma_{zt} & c^2\sigma_t^2 \end{bmatrix} \quad (20.37)$$

where σ^2 would come from a statistical analysis of the pseudorange errors. In this case, as the matrix terms of $(\mathbf{J}^T \mathbf{J})^{-1}$ term are completely dependent on the current LOS geometry of the transmitters to the receiver, \mathbf{J} is also known as the **geometry matrix**. It should be noted that this dependence implies the number of iterations to convergence depends on relative accuracy of LOS vectors.

One also analyze the relative precision of the pseudorange estimate due to the geometry of transmitter positions based on the diagonal terms of $(\mathbf{J}^T \mathbf{J})^{-1}$. These are called the **dilution of precision (DOP)** and for an NED reference frame for $x - y - z$, one can define the following DOPs:

- Vertical DOP (VDOP): σ_z
- Horizontal DOP (HDOP): $\sqrt{\sigma_x^2 + \sigma_y^2}$
- Position DOP (PDOP): $\sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2}$
- Geometric DOP (GDOP): $\sqrt{\sigma_x^2 + \sigma_y^2 + \sigma_z^2 + c^2 \sigma_t^2}$
- Time DOP (TDOP): σ_t

Differential Pseudorange Multilateration

Often one desires to form differential pseudoranges to obtain more accurate measurements for the positioning estimation. One common technique is to set the j^{th} transmitter as the **reference transmitter** with pseudorange equation

$$\rho_j = r_j + c(\delta t - \delta t_j) + b_{p,j} + b_{c,j} + \epsilon_j \quad (20.38)$$

Then $\forall i \neq j$, one can form the **between-transmitter single-difference (SD) pseudorange equation** with ρ_j as

$$\begin{aligned} \rho_i - \rho_j &= r_i + c\delta t - c\delta t_i + b_{p,i} + b_{c,i} + \epsilon_i \\ &\quad - r_j - c\delta t + c\delta t_j - b_{p,j} - b_{c,j} - \epsilon_j \end{aligned} \quad (20.39)$$

which can be rewritten succinctly by denoting the single-difference by the ∇_j , one has

$$\nabla_j \rho_i = \nabla_j r_i - c\nabla_j \delta t_i + \nabla_j b_{p,i} + \nabla_j b_{c,i} + \nabla_j \epsilon_i \quad (20.40)$$

which explicitly eliminates the receiver clock error which could be very noisy relative to the other errors in some systems.

Thus, one has for the parameter vector

$$\vec{\beta} = \vec{x} = \begin{bmatrix} x \\ y \\ z \end{bmatrix} \quad (20.41)$$

In addition, note that the between-receiver SD noise term for each single-difference is

$$\nabla_j \epsilon_i = \epsilon_i - \epsilon_j \quad (20.42)$$

where the j -dependent noise that will be common to all measurements, resulting in correlated measurements. If one assumes $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, the single-differencing with the reference transmitter produces correlated measurements with covariance matrix

$$\text{Cov}(\nabla_j \epsilon_i) = \sigma^2 \begin{bmatrix} 2 & 1 & \cdots & 1 & 1 \\ 1 & 2 & \cdots & 1 & 1 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 1 & 1 & \cdots & 2 & 1 \\ 1 & 1 & \cdots & 1 & 2 \end{bmatrix} \quad (20.43)$$

Furthermore, note that $\nabla_j \rho_i$ can be related to the time difference of arrival (TDOA) between signals i and j , $TDOA_{ij}$, by noting

$$\rho_i - \rho_j = c(TOA_i - TOT_i) - c(TOA_j - TOT_j) \quad (20.44)$$

Then, assuming $TOT_i - TOT_j \approx 0$, one can form the difference

$$\nabla_j \rho_i = c(TOA_i - TOA_j) = cTDOA_{ij} \quad (20.45)$$

Thus, this technique is useful if one can measure $TDOA_{ij}$ much more accurately than TOA_i and TOA_j individually, and one can setup the network of transmitters to transmit at the same time or know the differences with high precision, i.e. low $\nabla_j \delta t_i$. This technique is also known as **hyperbolic positioning** as one will obtain hyperbolic LOPs based on the differential ranges, r_{ij} .

The between-transmitter pseudorange equation allows one to form the nonlinear observation equation with $m + 1$ transmitters as

$$\vec{y} = \begin{bmatrix} \nabla_j \rho_1 \\ \vdots \\ \nabla_j \rho_m \end{bmatrix} = \mathbf{f}(\vec{x}, \hat{\beta}) = \begin{bmatrix} \|\hat{x} - \vec{x}_1\|_2 - \|\hat{x} - \vec{x}_j\|_2 \\ \vdots \\ \|\hat{x} - \vec{x}_m\|_2 - \|\hat{x} - \vec{x}_j\|_2 \end{bmatrix} \quad (20.46)$$

where $j = m + 1$ denotes the reference transmitter and \vec{x}_i denotes the position of the i^{th} transmitter.

To solve this problem using NLS with GNA, assume one has an initial estimate, $\hat{\beta}_0$. Then, one can iteratively improve $\hat{\beta}$ by $\vec{\Delta}_k$ for $k = 0, 1, \dots$, i.e.

$$\hat{\beta}_{k+1} = \hat{\beta}_k + \vec{\Delta}_k \quad (20.47)$$

where

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\vec{y} - \mathbf{f}(\vec{x}, \hat{\beta}_k)) \quad (20.48)$$

where the i^{th} row of the $m \times 3$ \mathbf{J} can be written as

$$\mathbf{J}_i = \left[\frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} - \frac{(\vec{x} - \vec{x}_j)^T}{\|\vec{x} - \vec{x}_j\|_2} \right]_{\vec{x} = \hat{x}_k} \quad (20.49)$$

Note that the single-difference with the reference transmitter produces correlated measurements and the estimator covariance is no longer well approximated by $\sigma^2 (\mathbf{J}^T \mathbf{J})^{-1}$.

Another technique for differential pseudorange multilateration considers the use of two or more receivers where one receiver is the **rover** whose position is to be estimated and may change over time, i.e. **kinematic**, while the other is the **base** whose position is *known* and typically is stationary, i.e. **static**. Thus, the position of the rover can be estimated by alternatively estimating the position of the rover relative to the base, also known as the **baseline**. Here, one can define the **rover pseudorange equation**, with dependent terms denoted by subscript R , and the i^{th} transmitter as

$$\rho_{i,R} = r_{i,R} + c(\delta t_R - \delta t_i) + b_{p,i} + b_{c,i,R} + \epsilon_{i,R} \quad (20.50)$$

Similarly, one can define the **base pseudorange equation**, with dependent terms denoted by subscript B , and the i^{th} transmitter as

$$\rho_{i,B} = r_{i,B} + c(\delta t_B - \delta t_i) + b_{p,i} + b_{c,i,B} + \epsilon_{i,B} \quad (20.51)$$

Then, one can form the **between-receiver single-difference (SD) pseudorange equation** between the rover and base as

$$\begin{aligned} \rho_{i,R} - \rho_{i,B} &= r_{i,R} + c(\delta t_R - \delta t_i) + b_{p,i} + b_{c,i,R} + \epsilon_{i,R} \\ &\quad - r_{i,B} - c(\delta t_B - \delta t_i) - b_{p,i} - b_{c,i,B} - \epsilon_{i,B} \end{aligned} \quad (20.52)$$

$$\begin{aligned} \rho_{i,R} - \rho_{i,B} &= r_{i,R} + c\delta t_R + b_{c,i,R} + \epsilon_{i,R} \\ &\quad - r_{i,B} - c\delta t_B - b_{c,i,B} - \epsilon_{i,B} \end{aligned} \quad (20.53)$$

which eliminates the transmitter clock and position errors exactly.

This equation can be rewritten succinctly by denoting the single difference between R and B terms with Δ_{RB} , i.e.

$$\Delta_{RB}\rho_i = \Delta_{RB}r_{i,RB} + c\Delta_{RB}\delta t + \Delta_{RB}b_{c,i} + \epsilon_i \quad (20.54)$$

Furthermore, assuming the rover and base are sufficiently close relative to geographical variation in the signal speed errors, one can reduce this equation to

$$\Delta_{RB}\rho_i = r_i + c\Delta_{RB}\delta t + \Delta_{RB}\epsilon_i \quad (20.55)$$

Thus, with between-receiver SD, one need not rely on accurate models for the known sources of error in the pseudoranges. However, the signal speed error increases as the baseline increases.

The parameter vector here is the rover baseline and the relative clock error between the rover and the base, $c\delta t_{RB}$, i.e.

$$\vec{\beta} = \begin{bmatrix} x_R - x_B \\ y_R - y_B \\ z_R - z_B \\ c\Delta_{RB}\delta t \end{bmatrix} \quad (20.56)$$

If one assumes a static base receiver, then the base position does not change and one can choose the reference frame such that $\vec{x}_B = [0 \ 0 \ 0]^T$ for relative positioning to the rover position \vec{x}_R . This results in the parameter vector

$$\vec{\beta} = \begin{bmatrix} x_R \\ y_R \\ z_R \\ c\Delta_{RB}\delta t \end{bmatrix} \quad (20.57)$$

The between-receiver single-differenced pseudorange equation allows one to form the nonlinear observation equation with m transmitters as

$$\vec{y} = \begin{bmatrix} \Delta_{RB}\rho_1 \\ \vdots \\ \Delta_{RB}\rho_m \end{bmatrix} = \mathbf{f}(\vec{x}, \hat{\vec{\beta}}) = \begin{bmatrix} \|\hat{\vec{x}}_R - \vec{x}_1\|_2 - \|\vec{x}_B - \vec{x}_1\|_2 + c\Delta_{RB}\delta t \\ \vdots \\ \|\hat{\vec{x}}_R - \vec{x}_m\|_2 - \|\vec{x}_B - \vec{x}_m\|_2 + c\Delta_{RB}\delta t \end{bmatrix} \quad (20.58)$$

If one assumes a static base receiver, then one can rewrite the nonlinear observation equation as

$$\mathbf{f}(\vec{x}, \hat{\beta}) = \begin{bmatrix} \|\hat{x}_R - \vec{x}_1\|_2 - \|\vec{x}_1\|_2 + c\Delta_{RB}\delta t \\ \vdots \\ \|\hat{x}_R - \vec{x}_m\|_2 - \|\vec{x}_m\|_2 + c\Delta_{RB}\delta t \end{bmatrix} \quad (20.59)$$

To solve this static base problem using NLS with GNA, assume one has an initial estimate, $\hat{\beta}_0$. Then, one can iteratively improve $\hat{\beta}_k$ by $\vec{\Delta}_k$ for $k = 0, 1, \dots$, i.e.

$$\hat{\beta}_{k+1} = \hat{\beta}_k + \vec{\Delta}_k \quad (20.60)$$

where

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\vec{y} - \mathbf{f}(\vec{x}, \hat{\beta}_k)) \quad (20.61)$$

where the i^{th} row of the $m \times 4$ \mathbf{J} can be written as

$$\mathbf{J}_i = \begin{bmatrix} \frac{(\vec{x}_R - \vec{x}_i)^T}{\|\vec{x}_R - \vec{x}_i\|_2} & 1 \end{bmatrix}_{\vec{x}_R = \hat{x}_{R,k}} \quad (20.62)$$

Lastly, another technique for differential pseudorange multilateration considers the between-transmitter differences for between-receiver differences which forms the **double-differenced (DD) pseudorange equation** as

$$\Delta_{RB}\rho_i - \Delta_{RB}\rho_j = \Delta_{RB}r_{i,RB} + c\Delta_{RB}\delta t + \Delta_{RB}\epsilon_i - \Delta_{RB}r_j - c\Delta_{RB}\delta t - \Delta_{RB}\epsilon_j \quad (20.63)$$

$$\Delta_{RB}\rho_i - \Delta_{RB}\rho_j = \Delta_{RB}r_i - \Delta_{RB}r_j + \epsilon_{i,RB} - \epsilon_{j,RB} \quad (20.64)$$

which can be rewritten succinctly as

$$\nabla_j \Delta_{RB}\rho_i = \nabla_j \Delta_{RB}r_i + \nabla_j \Delta_{RB}\epsilon_i \quad (20.65)$$

which eliminates the receiver clock error and leaves the double-differenced true range plus the double-differenced unmodeled errors. This technique can also reduce the contribution of the signal speed errors.

Here, one has for the parameter vector, the relative position of the rover to the base, i.e.

$$\vec{\beta} = \Delta_{RB} \vec{x} = \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} - \begin{bmatrix} x_B \\ y_B \\ z_B \end{bmatrix} \quad (20.66)$$

where $[x_B \ y_B \ z_B]^T$ is assumed known. If one assumes a static base receiver, then the base position does not change and one can choose the reference frame such that $\vec{x}_B = [0 \ 0 \ 0]^T$ for relative positioning to the rover position \vec{x}_R . This results in the parameter vector

$$\vec{\beta} = \vec{x}_R = \begin{bmatrix} x_R \\ y_R \\ z_R \end{bmatrix} \quad (20.67)$$

In addition, note that the double-differencing noise term for each double-difference is

$$\nabla_j \Delta_{RB} \epsilon_i = \epsilon_{i,R} - \epsilon_{i,B} - \epsilon_{j,R} + \epsilon_{j,B} \quad (20.68)$$

where the two j -dependent noise will be common to all measurements, results in correlated measurements. If one assumes $\epsilon \sim \mathcal{N}(0, \sigma^2 I)$, the double-differencing produces correlated measurements with covariance

$$\text{Cov}(\nabla_j \Delta_{RB} \epsilon_i) = \sigma^2 \begin{bmatrix} 4 & 2 & \cdots & 2 & 2 \\ 2 & 4 & \cdots & 2 & 2 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 2 & 2 & \cdots & 4 & 2 \\ 2 & 2 & \cdots & 2 & 4 \end{bmatrix} \quad (20.69)$$

The double-differenced pseudorange equation allows one to form the nonlinear observation equation with $m + 1$ transmitters as

$$\vec{\mathbf{y}} = \begin{bmatrix} \nabla_j \Delta_{RB} \rho_1 \\ \vdots \\ \nabla_j \Delta_{RB} \rho_m \end{bmatrix} = \mathbf{f}(\vec{\mathbf{x}}, \hat{\beta}) = \begin{bmatrix} (\|\hat{\vec{x}}_R - \vec{x}_1\|_2 - \|\vec{x}_B - \vec{x}_1\|_2) - (\|\hat{\vec{x}}_R - \vec{x}_j\|_2 - \|\vec{x}_B - \vec{x}_j\|_2) \\ \vdots \\ (\|\hat{\vec{x}}_R - \vec{x}_m\|_2 - \|\vec{x}_B - \vec{x}_m\|_2) - (\|\hat{\vec{x}}_R - \vec{x}_j\|_2 - \|\vec{x}_B - \vec{x}_j\|_2) \end{bmatrix} \quad (20.70)$$

If one again assumes a static base receiver, then one can rewrite the nonlinear observation equation as

$$\vec{\mathbf{y}} = \begin{bmatrix} \nabla_j \Delta_{RB} \rho_1 \\ \vdots \\ \nabla_j \Delta_{RB} \rho_m \end{bmatrix} = \mathbf{f}(\vec{\mathbf{x}}, \hat{\beta}) = \begin{bmatrix} (\|\vec{x}_R - \vec{x}_1\|_2 - \|\vec{x}_1\|_2) - (\|\vec{x}_R - \vec{x}_j\|_2 - \|\vec{x}_j\|_2) \\ \vdots \\ (\|\vec{x}_R - \vec{x}_m\|_2 - \|\vec{x}_m\|_2) - (\|\vec{x}_R - \vec{x}_j\|_2 - \|\vec{x}_j\|_2) \end{bmatrix} \quad (20.71)$$

To solve this static base problem using NLS with GNA, assume one has an initial estimate, $\hat{\beta}_0$. Then, one can iteratively improve $\hat{\beta}_k$ by $\vec{\Delta}_k$ for $k = 0, 1, \dots$, i.e.

$$\hat{\beta}_{k+1} = \hat{\beta}_k + \vec{\Delta}_k \quad (20.72)$$

where

$$\vec{\Delta}_k = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \hat{\beta}_k)) \quad (20.73)$$

where the i^{th} row of the $m \times 3$ \mathbf{J} can be written as

$$\mathbf{J}_i = \left[\frac{(\vec{x}_R - \vec{x}_i)^T}{\|\vec{x}_R - \vec{x}_i\|_2} - \frac{(\vec{x}_R - \vec{x}_j)^T}{\|\vec{x}_R - \vec{x}_j\|_2} \right]_{\vec{x}_R = \hat{\vec{x}}_{R,k}} \quad (20.74)$$

20.3 Phase-Based Positioning

Phase-Based Ranging

Another technique for positioning is to accurately measure the phase of a transmitted wave signal at a *constant* frequency. One complete cycle or wavelength, λ , of this signal is illustrated in the following graphic with different **phase angles** shown representing different points along the waveform. Assuming a receiver can track the phase of the wave signal in time, ϕ , as **fractional cycles**. Then, if the transmitter has created a very constant wavelength, a replicated wave signal in the receiver signal processing can be roughly matched in time to when the signal was transmitted, thus the fractional phase cycles correspond to a *distance* from the beginning of the current cycle defined as the wavelength multiplied by the fractional cycle, i.e. $\lambda\phi$. However, the portion of the signal that extends from the beginning of the currently measured cycle to the signal source, B , is unknown and, hence, is called the **phase ambiguity**.

Thus, the **phase-based range equation** for the i^{th} transmitter is defined as

$$\lambda\phi_i + B_i = r_i + c(\delta t - \delta t_i) + K + b_{p,i} + b_{c,i} + \epsilon_i \quad (20.75)$$

where k is the receiver **phase delay**, or **phase ambiguity** in its measurement and

$$B_i = K_i + \lambda N_i \quad (20.76)$$

where K_i is the transmitter **phase delay**, or *phase ambiguity*, and N_i is the **integer ambiguity**, i.e. the integer number of wavelengths from the transmitter to the beginning of the fractional cycles distance, $\lambda\phi$. It should be noted that K and K_i are also known as the **fractional ambiguities** as they can be any real number as opposed to N_i .

Furthermore, substituting for r_i and B_i , then by rearranging one has

$$\lambda\phi_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + c(\delta t - \delta t_i) + (K - K_i) - \lambda N_i + b_{p,i} + b_{c,i} + \epsilon_i \quad (20.77)$$

where it should be noted that rotating transmitters and receivers can also cause an additional additive **phase wind-up** error term, $\lambda\omega$, due to any angular velocity interacting with circularly polarized wave signals. In phase-based positioning, one may be able to refine the initially calculated $\lambda\phi_i$ based on highly accurate transmitter-dependent errors, δt_i , K_i , $b_{p,i}$, and $b_{c,i}$, one has

$$\lambda\phi_i = \sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2} + c\delta t + K - \lambda N_i + \epsilon_i \quad (20.78)$$

where x , y , z , δt , K , and N_i are the six unknown parameters.

Thus, in un-differenced phase-based positioning, one has for the parameter vector

$$\vec{\beta} = \begin{bmatrix} x \\ y \\ z \\ c\delta t \\ K \\ N_i \end{bmatrix} \quad (20.79)$$

where each transmitter will have an *unknown* integer ambiguity that must be estimated. This additional consideration of each N_i leads to additional estimation methods in phase-based positioning, namely estimating across multiple time steps as well as **ambiguity fixing**. The other potential difficulty in phase-based positioning is obtaining accurate enough models for the transmitter-dependent errors to be eliminated or reduced to a sufficient extent for estimating all N_i . Thus, the primary obstacles for un-differenced phase-based positioning are precise models for the transmitter position errors, clock errors, phase delays, and transmission speed errors which are typically only available in post-processing of phase data especially as the fractional ambiguities are not constant, but drift over time. Thus, differential phase-based positioning is typically used for real-time positioning and is discussed in this section.

Differential Phase-Based Positioning

Due to N_i appearing in the parameter vector, using a between-transmitter phase measurement will only increase the number of unknowns in phase-based positioning. Thus, for single-differencing one typically uses two or more receivers in a **rover** and **base** configuration with dependent terms distinguished by subscripts R and B , respectively. Thus, for the i^{th} transmitter, one has

$$\lambda\phi_{i,R} = r_{i,R} + c(\delta t_R - \delta t_i) + (K_R - K_i) - \lambda N_{i,R} + b_{p,i} + b_{c,i} + \epsilon \quad (20.80)$$

and

$$\lambda\phi_{i,B} = r_{i,B} + c(\delta t_B - \delta t_i) + (K_B - K_i) - \lambda N_{i,B} + b_{p,i} + b_{c,i} + \epsilon \quad (20.81)$$

Then, one can define the **single-differenced (SD) phase-based range equation** for each i^{th} transmitter as

$$\lambda\Delta_{RB}\phi_i = \Delta_{RB}r_i + c\Delta_{RB}\delta t + \Delta_{RB}K - \lambda\Delta_{RB}N_i + \Delta_{RB}\epsilon_i \quad (20.82)$$

where the term Δ_{RB} represents the difference of the corresponding rover term with the base term.

Thus, the SD operation eliminates the transmitter-dependent errors, i.e. δt_i , K_i , and $b_{p,i}$, as well as $b_{c,i}$ assuming a sufficiently small baseline. However, the differential clock and phase errors remain. Thus, one often sets the $j = m + 1$ as the reference transmitter, then $\forall i \neq j$, one can form the **double-difference (DD) phase-based range equation** for each i^{th} transmitter as

$$\lambda\nabla_j\Delta_{RB}\phi_i = \nabla_j\Delta_{RB}r_i - \lambda\nabla_j\Delta_{RB}N_i + \nabla_j\Delta_{RB}\epsilon_i \quad (20.83)$$

Thus, in double-differenced phase-based positioning, one has for the parameter vector

$$\vec{\beta} = \begin{bmatrix} x_R - x_B \\ y_R - y_B \\ z_R - z_B \\ \nabla_j\Delta_{RB}N_1 \\ \vdots \\ \nabla_j\Delta_{RB}N_m \end{bmatrix} \quad (20.84)$$

If one assumes a static base receiver, then the base position does not change and one can choose the reference frame such that $\vec{x}_B = [0 \ 0 \ 0]^T$ for relative positioning to the rover position \vec{x}_R . This results in the parameter

vector

$$\vec{\beta} = \begin{bmatrix} x_R \\ y_R \\ z_R \\ \nabla_j \Delta_{RB} N_1 \\ \vdots \\ \nabla_j \Delta_{RB} N_m \end{bmatrix} \quad (20.85)$$

one can see the DD phase measurements provide m equations and $m+3$ unknowns, i.e. the position of rover \vec{x}_R and the m integer ambiguities, $\nabla_j \Delta_{RB} N_i$, which does not allow a determinable solution at a *single* time step.

However, if the receiver can track the additional cycles from i^{th} transmitter, C_i , across multiple time steps. Then, as $\nabla_j \Delta_{RB} N_i$ are *constant* unknowns with respect to time, if four or more *of the same* transmitters are sampled at each subsequent time step, the number of unknowns will decrease since the *new* rover position has three unknown coordinates to consider. This concept requires a continuous lock on the signal in order to keep constant the initial integer number of unknown cycles, N . Thus, the rover and base receivers can alternatively measure the phase as a **fractional cycle count**, Φ_i . This concept is called **cycle counting**. With this alternative measurement, one has at time step k for the phase of the signal from the i^{th} transmitter

$$\Phi_{i,k} = \lambda \phi_{i,k} \quad (20.86)$$

and at time step $k+n$ for the phase of the signal from the i^{th} transmitter

$$\Phi_{i,k+n} = \lambda \phi_{i,k+n} = \lambda \phi_{i,k} + C_{i,n} \quad (20.87)$$

where $C_{i,n}$ is the fractional cycle count from k to $k+n$. Then, one can form the alternate **double-differenced cycle count range equation** for each i^{th} transmitter at time step k as

$$\nabla_j \Delta_{RB} \Phi_{i,k} = \nabla_j \Delta_{RB} r_{i,k} + \nabla_j \Delta_{RB} \lambda N_i + \nabla_j \Delta_{RB} \epsilon_{i,k} \quad (20.88)$$

where the sign of N has been changed without loss of generality due to the arbitrary start of the cycle counting.

Defining the following vectors for the $i = 1, \dots, m$ transmitters at time step k as

$$\vec{\Phi}_k = [\Phi_1 \ \dots \ \Phi_m]^T \quad (20.89)$$

$$\vec{r}_k = [r_1 \ \dots \ r_m]^T \quad (20.90)$$

$$\vec{N} = [N_1 \ \dots \ N_m]^T \quad (20.91)$$

$$\vec{\epsilon}_k = [\epsilon_1 \ \dots \ \epsilon_m]^T \quad (20.92)$$

one has the vector equation

$$\nabla_j \Delta_{RB} \vec{\Phi}_k = \nabla_j \Delta_{RB} \vec{r}_k + \lambda \nabla_j \Delta_{RB} \vec{N} + \nabla_j \Delta_{RB} \vec{\epsilon}_k \quad (20.93)$$

Then, for $k = 1, \dots, n$, one can form the stacked nonlinear observation equation as

$$\vec{y} = \begin{bmatrix} \nabla_j \Delta_{RB} \vec{\Phi}_1 \\ \vdots \\ \nabla_j \Delta_{RB} \vec{\Phi}_n \end{bmatrix} = \mathbf{f}(\vec{x}, \hat{\vec{\beta}}) = \begin{bmatrix} \left[\left(\|\vec{x}_{R,1} - \vec{x}_1\|_2 - \|\vec{x}_B - \vec{x}_1\|_2 \right) - \left(\|\vec{x}_{R,1} - \vec{x}_j\|_2 - \|\vec{x}_B - \vec{x}_j\|_2 \right) \right] \\ \vdots \\ \left[\left(\|\vec{x}_{R,1} - \vec{x}_m\|_2 - \|\vec{x}_B - \vec{x}_m\|_2 \right) - \left(\|\vec{x}_{R,1} - \vec{x}_j\|_2 - \|\vec{x}_B - \vec{x}_j\|_2 \right) \right] \\ \vdots \\ \left[\left(\|\vec{x}_{R,n} - \vec{x}_1\|_2 - \|\vec{x}_B - \vec{x}_1\|_2 \right) - \left(\|\vec{x}_{R,n} - \vec{x}_j\|_2 - \|\vec{x}_B - \vec{x}_j\|_2 \right) \right] \\ \vdots \\ \left[\left(\|\vec{x}_{R,n} - \vec{x}_m\|_2 - \|\vec{x}_B - \vec{x}_m\|_2 \right) - \left(\|\vec{x}_{R,n} - \vec{x}_j\|_2 - \|\vec{x}_B - \vec{x}_j\|_2 \right) \right] \\ + \lambda \begin{bmatrix} I_{(m) \times (m)} \\ \vdots \\ I_{(m) \times (m)} \end{bmatrix} [\nabla_j \Delta_{RB} \vec{N}] \end{bmatrix} \quad (20.94)$$

If one again assumes a static base receiver, then one can rewrite the nonlinear observation equation as

$$\mathbf{f}(\vec{x}, \hat{\vec{\beta}}) = \begin{bmatrix} \left[\left(\|\vec{x}_{R,1} - \vec{x}_1\|_2 - \|\vec{x}_1\|_2 \right) - \left(\|\vec{x}_{R,1} - \vec{x}_j\|_2 - \|\vec{x}_j\|_2 \right) \right] \\ \vdots \\ \left[\left(\|\vec{x}_{R,1} - \vec{x}_m\|_2 - \|\vec{x}_m\|_2 \right) - \left(\|\vec{x}_{R,1} - \vec{x}_j\|_2 - \|\vec{x}_j\|_2 \right) \right] \\ \vdots \\ \left[\left(\|\vec{x}_{R,n} - \vec{x}_1\|_2 - \|\vec{x}_1\|_2 \right) - \left(\|\vec{x}_{R,n} - \vec{x}_j\|_2 - \|\vec{x}_j\|_2 \right) \right] \\ \vdots \\ \left[\left(\|\vec{x}_{R,n} - \vec{x}_m\|_2 - \|\vec{x}_m\|_2 \right) - \left(\|\vec{x}_{R,n} - \vec{x}_j\|_2 - \|\vec{x}_j\|_2 \right) \right] \\ + \lambda \begin{bmatrix} I_{(m) \times (m)} \\ \vdots \\ I_{(m) \times (m)} \end{bmatrix} [\nabla_j \Delta_{RB} \vec{N}] \end{bmatrix} \quad (20.95)$$

where the entire parameter vector to be estimated is

$$\vec{\beta} = \begin{bmatrix} \vec{x}_{R,1} \\ \vdots \\ \vec{x}_{R,n} \\ \nabla_j \Delta_{RB} \vec{N} \end{bmatrix} \quad (20.96)$$

where $\vec{x}_{R,1}, \dots, \vec{x}_{R,n}$ are the rover positions from $k = 1, \dots, n$, and each transmitter position, $\vec{x}_1, \dots, \vec{x}_m, \vec{x}_j$, may also change with k , though not explicitly written above.

To solve this static base problem using NLS with GNA, assume one has an initial estimate, $\hat{\beta}_0$. Then, one can iteratively improve $\hat{\beta}_p$ by Δ_p for $p = 1, \dots$

$$\vec{\Delta}_p = (\mathbf{J}^T \mathbf{J})^{-1} \mathbf{J}^T (\vec{\mathbf{y}} - \mathbf{f}(\vec{\mathbf{x}}, \hat{\beta}_p)) \quad (20.97)$$

where

$$\mathbf{J} = \begin{bmatrix} \mathbf{J}_1 & 0 & \cdots & \cdots & 0 & \lambda I_{m \times m} \\ 0 & \ddots & \ddots & \ddots & 0 & \lambda I_{m \times m} \\ 0 & \ddots & \mathbf{J}_k & \ddots & 0 & \lambda I_{m \times m} \\ 0 & \ddots & \ddots & \ddots & 0 & \lambda I_{m \times m} \\ 0 & \cdots & \cdots & 0 & \mathbf{J}_n & \lambda I_{m \times m} \end{bmatrix} \quad (20.98)$$

and the i^{th} row of each $m \times 3$ \mathbf{J}_k matrix can be written as

$$\mathbf{J}_{k,i} = \left[\frac{\vec{x}_{R,k} - \vec{x}_i}{\|\vec{x}_{R,k} - \vec{x}_i\|_2} - \frac{\vec{x}_{R,k} - \vec{x}_j}{\|\vec{x}_{R,k} - \vec{x}_j\|_2} \right]^T \vec{x}_{R,k} = \hat{\vec{x}}_{R,k,p} \quad (20.99)$$

which provides the estimated rover positions at each time step k as $\hat{\vec{x}}_{R,k}$, and the estimated $\nabla_j \Delta_{RB} \hat{N}_i$ as “real” numbers, a solution known as the **float solution** due to the computer science term for decimal numbers in algorithms. This is an imprecise estimate due to the fact that $\nabla_j \Delta_{RB} \hat{N}_i$ should be constrained to be integers.

Integer Ambiguity Fixing

To obtain the most accurate positioning solution beyond the NLS estimate, regardless whether using undifferenced or double-differenced phase measurements, one must “fix” the integer ambiguities as integers, i.e. obtain a **fixed solution** for which there are three primary methods with the simplest as rounding the float solution to the nearest integers, i.e. the **rounding** method.

A second method is to select p “candidate sets” of integers “near” the float solution, i.e. $\vec{N}_s \in \mathbb{S}$ with $s = 1, \dots, p$. Then, compute the positioning solution for each candidate set as

$$\hat{\vec{x}}_{1:n, \vec{N}_s} = [\hat{\vec{x}}_1, \dots, \hat{\vec{x}}_n] \quad \text{assuming } \nabla_j \Delta_{RB} \vec{N} = \vec{N}_s \quad (20.100)$$

for $s = 1, \dots, p$. Finally, the optimal fixed solution is chosen as the candidate set which minimizes the smallest post-fit residual between the measurements and the expected measurement based on the estimated position for that candidate set, i.e.

$$\hat{N} = \min_{\vec{N}_s \in \mathbb{S}} \left\| \vec{\mathbf{y}} - \mathbf{f} \left(\vec{\mathbf{x}}, [\hat{\vec{x}}_{1:n, \vec{N}_s}, \vec{N}_s]^T \right) \right\|_2 \quad (20.101)$$

It should be noted that this method requires possibly computing the positioning solution for a large number of candidate sets which can be computationally expensive if the number of measurements and time steps is large. Thus, this method is typically called the **brute force** method if care is not taken when selecting the candidate sets. An optimal way to choose a small set of integer candidates is to use the **Least-squares AMbiguity Decorrelation Adjustment (LAMBDA)** method which identifies the most likely candidates

using the geometry matrix and the correlation between the measurements to find the most likely candidate sets from a statistical perspective. Implementations of the LAMBDA method are available for free at www.tudelft.nl.

A third method uses an additionally available pseudorange measurement from all transmitters, i.e. $\vec{\rho}_k$ for $k = 1, \dots, n$, uses the integers which minimize the residual between phase-based measurement and the *expected* phase measurements based on the pseudorange solution. This is called the **geometry-free** method as its measurement model does not use the LOS vectors, i.e. the geometry matrix, for \hat{N} . For the double-differenced phases and *expected* phases based on the double-differenced pseudoranges, i.e.

$$\nabla_j \Delta_{RB} \hat{N} = \text{round} \left(\begin{bmatrix} \nabla_j \Delta_{RB} \vec{\Phi}_1 \\ \vdots \\ \nabla_j \Delta_{RB} \vec{\phi}_n \end{bmatrix} - \frac{1}{\lambda} \begin{bmatrix} \nabla_j \Delta_{RB} \vec{\rho}_1 \\ \vdots \\ \nabla_j \Delta_{RB} \vec{\rho}_n \end{bmatrix} \right) \quad (20.102)$$

Cycle Slip and Triple-Differencing

A **cycle slip** is a discontinuity in the receiver's signal processor digital phase-lock loop (DPLL) due to a loss in tracking a transmitter's signal. Cycle slip can be caused by several different factors including power loss, a failure of the receiver software, a malfunctioning clock oscillator, and most commonly, obstructions to the LOS to the transmitter such that the signal is lost. However, once the transmitter reappears, the reacquisition of the signal occurs and the DPLL can continue to count the cycles. The precision of the phase measurement requires an accurate counting of the cycles that have occurred to maintain a fixed integer ambiguity required for phase-based positioning.

However, in many applications one may desire to fix cycle slips by estimating the missing number using the other transmitter measurements to estimate the discrepancy. This type of analysis is often done by first using an additional **between-epochs difference** to form the **triple-difference cycle count range equation**, also known as the **transmitter-receiver-time triple-difference**. This technique takes the double-differenced phase-based range equation at different time steps k , i.e.

$$\lambda \nabla_j \Delta_{RB} \vec{\phi}_k = \nabla_j \Delta_{RB} \vec{r}_k - \lambda \nabla_j \Delta_{RB} \vec{N} + \nabla_j \Delta_{RB} \vec{\epsilon}_k \quad (20.103)$$

and computes the difference with one of the time steps, e.g. differencing $k = 1, \dots, n$ with $k = 0$, to form

$$\Delta_0 \nabla_j \Delta_{RB} \vec{\phi}_k = \nabla_j \Delta_{RB} \vec{\phi}_k - \nabla_j \Delta_{RB} \vec{\phi}_0 \quad (20.104)$$

$$\lambda \Delta_0 \nabla_j \Delta_{RB} \vec{\phi}_k = \Delta_0 \nabla_j \Delta_{RB} \vec{r}_k + \Delta_0 \nabla_j \Delta_{RB} \vec{\epsilon}_k \quad (20.105)$$

However, as before, though this eliminates a common known error, it increases the effect of the noise ϵ . Thus, triple-differencing is not as useful as double-differenced phase-based solutions for positioning. However, because it is less sensitive to cycle slips, it allows for corrections to be made through residual analysis of the component double-differences of the triple-differenced solution.

20.4 GNSS-Based Positioning

GNSS Code Phase Positioning

GNSS uses modulated radio signals made up of a central carrier frequency and encodes digital code through binary code modulation (BCM) onto the carrier signal. The original design and standard method of using

GNSS is reading and interpreting this digital code to form pseudorange measurements for positioning, a technique also known as the **code phase positioning** for GNSS. Using its own clock, GNSS receiver measures the time of arrival at the receiver, TOA_i , of the GPS code phase signal from the i^{th} satellite which has an encoded time of transmission, TOT_i . Thus, one can form the GNSS pseudorange measurement by

$$\rho_i = c (TOA_i - TOT_i) \quad (20.106)$$

where the GNSS timing is based off the fundamental clock rate, F_0 . For the **GNSS pseudorange equation**, one can relate the pseudorange ρ_i to the true range r_i between the receiver and the i^{th} satellite as

$$\rho_i = r_i + c (\delta t - \delta t_i) + b_{E,i} + b_{I,i} + b_{T,i} + \epsilon_{M-P} + \epsilon \quad (20.107)$$

where ϵ is the random unknown error sources, and $c\delta t$, $c\delta t_i$, $b_{E,i}$, $b_{I,i}$, $b_{T,i}$, and ϵ_{M-P} are the known error sources to ρ_i , collectively known as the **User Equivalent Range Error (UERE)** in GNSS applications.

The individual terms of the UERE and the source of their error can be summarized as follows. The unknown δt results from the receiver clock error due to an imperfect oscillator and must be estimated alongside the position estimate. The control segments and IGS monitor the **satellite clock error**, δt_i , as well as the **satellite ephemeris error**, $b_{E,i}$, and provide models to remove this error. Both of these terms can be improved through optimal smoothing *after* real-time observations of the GNSS signals. The signal speed transmission errors are primarily due to the **ionospheric delay**, $b_{I,i}$, and the **tropospheric delay**, $b_{T,i}$, depend on the atmospheric conditions and are collectively known as the **atmospheric delay**. Weather stations around the world provide regional information about the measured atmospheric delay, but typically these model improvements are made *after* real-time observations. Lastly, multi-path errors, ϵ_{M-P} , occur when the GNSS signal strongly reflects off a nearby object before reaching the receiver, thereby negating the assumption of straight flight from the transmitter to the receiver.

GNSS Carrier Phase Positioning

An alternative to the code phase positioning is **carrier phase positioning**. The primary reason that using the GNSS carrier phase in a phase-based positioning algorithm is the fact that it offers increased precision of range due to the fact that its wavelength is much shorter than the modulated encoded signal. A convenient approximation for the maximum resolution of the encoded and carrier signals is roughly one percent of the signal wavelength, also known as the **one percent rule**. For reference, the C/A code phase has a frequency f of 1.023 MHz which corresponds to a wavelength λ of 293 m by $\lambda = \frac{c}{f}$ where $c \approx 2.998 \times 10^8$ m/s. Then, by the one percent rule, this corresponds to a resolution of approximately ± 3 m. In practice, C/A positioning offers around ± 20 m of accuracy. Conversely, the L1 and L2 carrier signals have frequencies of 1575.42 and 1227.60 MHz which translates to 19 and 24 cm wavelengths. This effectively puts the one percent rule resolution of the carrier phase measurements as ± 2 mm.

Similar to the code phase, the **GNSS carrier phase range equation**, one can relate the carrier phase fractional cycle count, Φ_i , to the true range, r_i , between the receiver and the i^{th} satellite as

$$\Phi_i = \lambda\phi_i = r_i + c (\delta t - \delta t_i) + (K - K_i) + \lambda N_i + b_{E,i} - b_{I,i} + b_{T,i} + \lambda\omega + \epsilon_{M-P} + \epsilon \quad (20.108)$$

where ϵ is the random unknown error sources, and $c\delta t$, $c\delta t_i$, K , K_i , $b_{E,i}$, $b_{I,i}$, $b_{T,i}$, and ϵ_{M-P} are the carrier phase **user equivalent range error (UERE)**. Note that the ionosphere error is negative for the carrier phase measurement.

Also, the additional $\lambda\omega$ is known as the **phase wind-up** due to the circular polarization of the GNSS signal. This wind-up can often be estimated using the **geometry-free combination** for the carrier phase as

$$\Phi_{G-F} = \Phi_1 - \Phi_2 \quad (20.109)$$

where Φ_1 and Φ_2 are GNSS cycle counts on different frequencies. It should also be noted that one can form the **wide-lane combination** for the carrier phase as

$$\Phi_{W-L} = \frac{f_1\Phi_1 - f_2\Phi_2}{f_1 - f_2} \quad (20.110)$$

where Φ_1 , and Φ_2 are GNSS cycle counts on frequencies, f_1 and f_2 , respectively, which allows the integer ambiguities to be increased up to four times larger for the wavelength distance which makes the ambiguity fixing easier to estimate.

GNSS un-differenced carrier phase positioning is known as **precise point positioning (PPP)**. In order to remove as much error as possible, PPP requires error models for the precise transmitter orbits and clocks, the atmospheric effects, the antenna biases and orientation, and solid Earth tides with optional ocean loading and pole tides. Normally, the IGS data products for these errors are used due to their high accuracy. PPP also uses dual-frequency GNSS to remove the ionospheric errors. The PPP algorithm uses both carrier phase and pseudorange measurements across time in a nonlinear Kalman filter to estimate the receiver position and the receiver clock bias as well as the m transmitter combined fractional and integer phase ambiguities, and the wet tropospheric delay. However, a large convergence time is required for PPP filters to resolve the ambiguity fix, typically on the order of tens of minutes. Thus, PPP is primarily used as a post-processed positioning technique as an alternative to GNSS double-differenced carrier phase positioning as PPP allows a ubiquitous solution as no baseline limitations apply for the positioning.

GNSS double-differenced carrier phase positioning is known as **real-time kinematic (RTK)** and **post-processed kinematic (PPK)** depending on the time when the positioning is performed. The simplest form of RTK uses a base station with a radio data link to broadcast the raw observational data, and/or other correctional data, at about a 1 Hz rate while including its ECEF location every 10-30 seconds which allows rovers to compute a double-differenced NLS solution. An open protocol for general-purpose RTK is the **Radio Technical Commission for Maritime Services (RTCM)** standard. It should be noted that as RTK requires a data link to transmit the information to the rovers which it must first calculate, there will typically be a latency to the RTK communication. Thus, RTK receivers must send a range rate correction along with their data at the time of their observations for the rovers to apply the observations correctly. This radio data link also requires that a communication frequency is available that is not being used by other RTK users in the area.

Alternatively, **real-time networks (RTN)** use an internet connection to supply the corrections and typically received through the standard Networked Transport of RTCM via Internet Protocol (NTRIP) message using a cell phone modem. With this configuration, the corrections can be supplied by a network of stationary GNSS receivers, e.g. **continuous operating reference stations (CORS)**, which compute the GNSS distance-dependent biases, i.e. ephemeris, ionospheric, and tropospheric, over a wide area and compute corrections for users within the network area. One common method is the use of a **virtual reference station (VRS)** for which the rover receives the optimal corrections. Then, if a rover moves significantly away from the VRS, e.g. 10 km, a new VRS is provided to the rover. However, VRS requires feedback to the RTN to provide an approximate position. This is typically handled as a **National Marine Electronics**

Association (NMEA) standard string format, of which there are seven. It should also be noted that CORS networks can also be used for PPK as well as they provide their raw GNSS observables for public use.

User Equivalent Range Error Models

The satellite clock error models include a clock offset, a clock drift model, as well as general and special relativity, i.e.

$$\delta t_i = a_0 + a_1(t - t_0) + a_2(t - t_0)^2 + \Delta_{rel} \quad (20.111)$$

where a_0 is the **satellite clock offset**, a_1 is the **satellite clock drift**, a_2 is the **satellite clock drift rate**, and Δ_{rel} is the relativistic error term. This is typically modeled as

$$\Delta_{rel} = -2 \frac{\vec{r}_s \vec{v}_s}{c^2} = -2 \frac{\sqrt{\mu a}}{c^2} e \sin(E) \quad (20.112)$$

where \vec{r}_s is the radial position of the satellite, \vec{v}_s is the velocity of the satellite, c is the speed of light in a vacuum, μ is the geocentric gravitational constant, a is the semimajor axis, e is the eccentricity (about 0.02), and E is the eccentric anomaly. This effect can cause a pseudorange error of about 14 m. In addition, there is an additional effect due to the motion of the receiver on the Earth's surface as the earth rotates during transmission time. This is known as the *Sagnac effect* and can result in around 133 nanoseconds of error at its maximum.

As part of GNSS, the control segments monitor all satellite systems and broadcasted signals, thus making it capable of monitoring and controlling the satellite clocks which it guides to be in line with the GNSS time. Similar to the receiver clocks, the satellite clock are subject to destabilizing effects, e.g. temperature, satellite acceleration, radiation. However, the control segment must keep the satellite clocks within strict standards of GNSS time and publishes the current satellite clock error models through the broadcast message onboard the satellites. Though the control segments can monitor the clocks closely, they do not constantly tweak the satellite clocks since doing so would cause them to deteriorate much more rapidly. As such, clock failure is the most common failure mode currently. Thus, as these models are not completely accurate, but the remainder of the error can be treated as part of the other noise/uncertainty in the pseudorange, ϵ .

Other forces and moments than Newtonian gravity act on the satellites due to the non-uniformity of the Earth's gravity, the Sun's gravity, the Moon's gravity, and solar radiation pressure. However, each of these error sources are difficult to directly model the orbital errors analytically. Thus, the best model of these disturbing forces and moments is by real-time estimation of the current satellite motion. These estimates are then broadcast by control segment to the satellites for use in the navigation message. The real-time parameters are updated every 1-4 hours, but are relatively accurate. However, due to this real-time modeling, one can also post-process the GNSS data to form more optimized orbits for the satellites for post-processed positioning. Three publicly available orbital error products are published by the International GNSS Service (IGS) and are named by GPS Week number and Day of Week. These are the **ultra-rapid orbital errors** (iguWWWD.sp3) which has a 6-hour latency release and uses a constrained optimization technique that does not allow any net rotation or translation, the **rapid** (igrWWWD.sp3) which has a 13-hour latency release and uses a constrained optimization technique that also does not allow any net rotation or translation, and the **final** (igsWWWD.sp3) which has a 12 to 14 day latency release and uses a minimally constrained optimization that only does not allow any net rotation.

The electromagnetic properties of the GNSS signal changes as it passes through the Earth's **atmosphere** which can be modeled as four regimes. Namely, the ionosphere which spans roughly 50-1000 km, the mesosphere which spans roughly 40-60 km, stratosphere which spans roughly 15-40 km, troposphere which spans roughly 0-15 km. Through refraction and diffraction, the atmosphere alters the apparent speed and direction of the GNSS signal which causes an apparent delay in the signal transit. This atmospheric delay is primarily caused by the physics of the ionosphere and troposphere whose error contributions to the GNSS range equations can be modeled as follows.

When gas molecules are ionized by the Sun's ultraviolet radiation, free electrons are released into the atmosphere. These free electrons gather into the ionosphere, but their number and dispersion varies across time and geography and affect the electron density in ionosphere. This phenomenon can be quantified by the **total electron count (TEC)** whose TEC units (TECU) are multiples of 10^{16} free electrons in a vertical ionospheric column with a cross-sectional area of 1 m^2 , i.e. $1\text{TECU} = 10^{16} e^-/\text{m}^2$. The TEC depends on the time of day primarily through the day/night cycle, 11-year solar cycle, magnetic activity of the Earth, and geographic location. The highest TEC values and variations generally occur in a band of 60° of latitude around magnetic equator, but can also be harsh in the polar regions.

As the free electrons in the ionosphere dispersively slows the signal, there are frequency and path of transmission dependencies for the ionospheric delay. With these considerations, it has been shown that the first order ionospheric delay is proportional to the **slant total electron count (STEC)**, i.e. the total count along the length of transmission distance, l , i.e.

$$STEC = \int N_e dl \quad (20.113)$$

where N_e is the electron density. This effects can be described by the heuristic equation

$$b_I = \frac{40.3 \times 10^{16}}{f^2} STEC \quad (20.114)$$

where f is the frequency in Hz, and the $STEC$ is in TECUs. However, as some ionospheric delay may remain, one often uses a **mask angle** for elevations of 15° , i.e. satellites with elevations below 15° are not used in an un-differenced positioning solution.

Alternatively, for GNSS broadcast at multiple frequencies, e.g. GPS, one can use a multi-frequency receiver to form an **ionosphere-free combination** using two frequencies for the ionosphere-free pseudorange, ρ_{I-F} , as

$$\rho_{I-F} = \frac{f_1^2 \rho_1 - f_2^2 \rho_2}{f_1^2 - f_2^2} \quad (20.115)$$

or the ionosphere-free cycle count, Φ_{I-F} , as

$$\Phi_{I-F} = \frac{f_1^2 \Phi_1 - f_2^2 \Phi_2}{f_1^2 - f_2^2} \quad (20.116)$$

where ρ_1 , ρ_2 , Φ_1 , and Φ_2 are GNSS pseudoranges and cycle counts on frequencies, f_1 and f_2 , respectively.

The tropospheric delay consists of two components which cause refraction of the GNSS signal: the **hydrostatic delay**, also known as the **dry delay**, and the **wet delay**. The hydrostatic delay causes 80-90% of the total tropospheric delay and is closely correlated to the atmospheric pressure, thus easily estimated.

The wet delay is due to the highly variable water vapor distribution which is only roughly correlated to temperature and humidity. Thus, some high-precision GNSS applications may use water vapor radiometers and radiosondes to better model this effect. Modeling the troposphere delay can be up to 95% effective, but unlike the ionospheric effect it is nondispersive, i.e. it affects all frequencies the same, and cannot be removed using a dual-frequency receiver. However, it is more consistent than the ionosphere delay. It is also known that the index of refraction decreases with elevation angle ϵ . At MSL a 90° elevation only produces about 2.5 m of error in the pseudorange, while at 75° elevation, this would be roughly 9 m of error, and at 10° elevation this would be 20 m of error. Thus, using a mask angle can also be useful for removing a significant portion of the tropospheric delay.

Therefore, one typically uses a model of the form:

$$b_T = m_h(\epsilon)ZHD + m_d(\epsilon)ZWD \quad (20.117)$$

where $m_h(\epsilon)$ and $m_d(\epsilon)$ are the Niell mapping functions based on ϵ , ZHD is the zenith hydrostatic delay, and ZWD is the zenith wet delay. ZHD and ZWD related to the air total pressure, p (hPa), the temperature, T (K), and the water vapor pressure, e (hPa). Two common models for these are

$$ZHD = \frac{0.0022767p}{1 - 0.00266 \cos 2\phi - 0.00028h_0} \quad (20.118)$$

where ϕ is the ellipsoidal latitude, h_0 is the surface height above ellipsoid (m), p is the total surface pressure (hPa), and

$$ZWD = 10^{-3} \int_{h_0}^{\infty} \left(22.1 \frac{e}{T} + 3.754 \frac{e}{T^2} \right) \quad (20.119)$$

Multi-path occurs when part of the GNSS signal broadcast from the satellite reaches the receiver antenna after one or more reflections which can be caused by objects such as the ground, buildings, cliffs, and trees. These reflections are still sensed by the antenna and interfere with direct signal by skewing the correlation peak during the replica code matching done for acquiring the PRN code and obtaining a time difference measurement. This skewing can be up to 1.5 code chip lengths. However, once a replica PRN code is correlated with the incoming signal, other multi-path signals outside the expected chip length are easily rejected by modern receivers. However, smaller multi-path delays less than the expected chip length are harder to filter out and can result in typically 1-3 meters of additional error to the pseudorange measurement. One method for further reducing multi-path is to use the fact that for an odd number of reflections, the multi-path GNSS signals become **left hand circular polarized** which allow advanced receivers to identify and remove. Multi-path effects can also practically be lessened by using a mask angle for elevations of 15°, elevating the antenna, and using a plate to block signals from below.

Introductory Navigation Systems

21.1 Introduction to Navigation Systems

Navigation systems are computing systems that provide estimates of the position of a vehicle relative to some reference frame while *in motion*. For Earth-based navigation, one typically uses the LLA coordinates of the ECEF reference frame to navigate from one point on the Earth's surface to another. For space-based navigation, one typically uses a planet-, moon-, or sun-centered inertial reference frame to navigate in orbit. Thus, positioning systems can serve as navigation systems, but navigation systems can also sense the motion of a vehicle to infer the position of a moving vehicle, a process known as **dead reckoning**.

Some dead reckoning navigation systems use the vehicle's distance traveled directly in the position update. A sensor that measures distance traveled is known as an **odometer**. While odometers for ground vehicles can be based on wheel rotations, odometers for aerospace vehicles can be developed based on 2D or 3D imagers from electro-optical sensors which estimate the change in the vehicle's position from one image frame to the next using auto-correlation techniques between subsequent images.

For a odometry-based dead reckoning system, the LLA position in the ECEF frame can be predicted by the model

$$\vec{p}_{E,k} = \begin{bmatrix} \dot{\ell}_k \\ \dot{\lambda}_k \\ \dot{h}_k \end{bmatrix} = \begin{bmatrix} \dot{\ell}_{k-1} \\ \dot{\lambda}_{k-1} \\ \dot{h}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{1}{R_N(\ell_{k-1})+h_{k-1}} & 0 & 0 \\ 0 & \frac{1}{(R_E(\ell_{k-1})+h_{k-1}) \cos \ell_{k-1}} & 0 \\ 0 & 0 & -1 \end{bmatrix} \vec{u}_{N,k} \quad (21.1)$$

where $\vec{u}_{N,k}$ is the change in position from $k-1$ to k as measured in the navigation frame. Notably in this case, an approximation of the velocity at time step k , $\vec{v}_{N,k}$, is given simply by

$$\vec{v}_{N,k} \approx \vec{u}_{N,k} \Delta t \quad (21.2)$$

where Δt is the sampling time interval of the odometer.

Some dead reckoning navigation systems use velocity directly in the position update. A sensor that measures velocity is known as a **velocimeter**. Electro-optical velocimeters use Doppler frequency measurements from electromagnetic signals to infer relative range rate from features in the environment. In this case, one

requires the pre-processing of multiple **pseudorange rates** using a least-squares estimator for the velocity measurement. Another approach for aircraft is to use the airspeed vector from an air data system (ADS) as a rough approximation for the inertial velocity, i.e. as no-wind and flat-Earth approximations, which may or may not include subtracting a provided wind speed from the provided airspeed.

For a velocimetry-based dead reckoning system, the LLA position in the ECEF frame is related to the velocity kinematically through the ordinary differential equation

$$\dot{\vec{p}}_E = \begin{bmatrix} \dot{\ell} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_N+h} & 0 & 0 \\ 0 & \frac{1}{(R_E+h)\cos\ell} & 0 \\ 0 & 0 & -1 \end{bmatrix} \vec{v}_N \quad (21.3)$$

where \vec{v}_N is the instantaneous velocity of the vehicle in the navigation frame. This model can be integrated using an Euler integration to obtain a position prediction model as

$$\vec{p}_{E,k} = \begin{bmatrix} \dot{\ell}_k \\ \dot{\lambda}_k \\ \dot{h}_k \end{bmatrix} = \begin{bmatrix} \dot{\ell}_{k-1} \\ \dot{\lambda}_{k-1} \\ \dot{h}_{k-1} \end{bmatrix} + \Delta t \begin{bmatrix} \frac{1}{R_N(\ell_{k-1})+h_{k-1}} & 0 & 0 \\ 0 & \frac{1}{(R_E(\ell_{k-1})+h_{k-1})\cos\ell_{k-1}} & 0 \\ 0 & 0 & -1 \end{bmatrix} \vec{v}_{N,k} \quad (21.4)$$

where Δt is the sampling time interval of the velocimeter. For an accelerometry-based dead reckoning system, the velocity is first updated as

$$\vec{v}_{N,k} = \vec{v}_{N,k-1} + \Delta t \vec{a}_{N,k-1} \quad (21.5)$$

where $\vec{a}_{N,k-1}$ is the instantaneous velocity of the vehicle in the navigation frame and Δt is the sampling time interval of the accelerometer.

Lastly, it should be noted that most odometers, velocimeters, and accelerometers are **strapdown sensors**, i.e., the sensors are attached to the structure of the vehicle and their measurements are referenced to the body-fixed frame of the vehicle. In this case, the navigation system will also require attitude determination to transform the velocity vector to the navigation frame for the dead reckoning computation. Thus, many navigation systems use a combination of attitude determination and dead reckoning equations which implicitly provides estimates of the position, velocity, and attitude (PVA) of the vehicle.

Electro-Optical Odometry

Doppler Positioning and Velocimetry

The true range rate, \dot{r}_i , for the i^{th} transmitter is related to the velocity of the receiver, $\vec{x} = [\dot{x} \ \dot{y} \ \dot{z}]^T$, and the velocity of the i^{th} transmitter, $\vec{x}_i = [\dot{x}_i \ \dot{y}_i \ \dot{z}_i]^T$, through the relative line-of-sight (LOS) vector, i.e.

$$\dot{r}_i = (\vec{x} - \vec{x}_i)^T \frac{(\vec{x} - \vec{x}_i)}{\|\vec{x} - \vec{x}_i\|_2} = \frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} (\vec{x} - \vec{x}_i) \quad (21.6)$$

or

$$\dot{r}_i = \frac{(x - x_i)(\dot{x} - \dot{x}_i) + (y - y_i)(\dot{y} - \dot{y}_i) + (z - z_i)(\dot{z} - \dot{z}_i)}{\sqrt{(x - x_i)^2 + (y - y_i)^2 + (z - z_i)^2}} \quad (21.7)$$

where $\vec{x} = [x \ y \ z]^T$ is the receiver position and $\vec{x}_i = [x_i \ y_i \ z_i]^T$ is the i^{th} transmitter's position.

The **pseudorange rate equation**, one can relate the pseudorange rate, $\dot{\rho}_i$, to the true range rate, \dot{r}_i , between the receiver and the i^{th} transmitter as

$$\dot{\rho}_i = \dot{r}_i + c\dot{t} + \epsilon \quad (21.8)$$

where \dot{t} is the **receiver clock drift** and accounts for the timing drift in the rate measurement. Lastly, ϵ accounts for any additional unknown random errors, e.g. electrical noise.

A energy wave receiver typically measures the **Doppler shift** at the receiver, Δf_i , of a *known* signal from the i^{th} transmitter with *known* frequency of transmission, f . Thus, one can form the **pseudorange rate measurement**, $\dot{\rho}_i$, for the i^{th} transmitter by

$$\dot{\rho}_i = c \frac{\Delta f_i}{f} \quad (21.9)$$

Thus, by substitution, one has the **Doppler positioning equation** which relates the Doppler shift, Δf_i

$$\Delta f_i = \frac{f}{c} \frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} (\dot{\vec{x}} - \dot{\vec{x}}_i) + f\dot{t} + \epsilon \quad (21.10)$$

which depends on seven unknowns, receiver position, velocity, and clock drift. One may also include bias errors due to the assumed transmitter positions and the signal transmission speed and frequency.

Thus, in pseudorange rate navigation, one has for the parameter vector

$$\vec{\beta} = \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \\ f\dot{t} \end{bmatrix} \quad (21.11)$$

which implies that **Doppler positioning** requires seven or more transmitters to be able to obtain a position *and* velocity estimate. As such, Doppler positioning is more often used for *static* receivers, so only four transmitters required. The Doppler range equation allows one to form a nonlinear observation equation with m transmitters as

$$\vec{y} = \begin{bmatrix} \Delta f_1 \\ \vdots \\ \Delta f_m \end{bmatrix} = \mathbf{f}(\vec{x}, \hat{\vec{\beta}}) = \begin{bmatrix} \frac{f}{c} \frac{(\vec{x} - \vec{x}_1)^T}{\|\vec{x} - \vec{x}_1\|_2} (\dot{\vec{x}} - \dot{\vec{x}}_1) + f\dot{t} \\ \vdots \\ \frac{f}{c} \frac{(\vec{x} - \vec{x}_m)^T}{\|\vec{x} - \vec{x}_m\|_2} (\dot{\vec{x}} - \dot{\vec{x}}_m) + f\dot{t} \end{bmatrix} \quad (21.12)$$

To solve this problem using NLS with GNA, assume one has an initial estimate, $\hat{\vec{\beta}}_0$. Then, one can iteratively improve $\hat{\vec{\beta}}$ by $\vec{\Delta}_k$ for $k = 0, 1, \dots$ as

$$\hat{\vec{\beta}}_{k+1} = \hat{\vec{\beta}}_k + \vec{\Delta}_k \quad (21.13)$$

where

$$\vec{\Delta}_k = \left(\mathbf{J}^T \mathbf{J} \right)^{-1} \mathbf{J}^T \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\vec{\beta}}_k) \right) \quad (21.14)$$

where the i^{th} row of $m \times 7 \mathbf{J}$ can be written as

$$\mathbf{J}_i = \begin{bmatrix} \frac{f}{c} \left(\frac{(\vec{x} - \vec{x}_i)}{\|\vec{x} - \vec{x}_i\|_2} - (\vec{x} - \vec{x}_i) (\dot{\vec{x}} - \dot{\vec{x}}_i)^T \frac{(\vec{x} - \vec{x}_i)}{\|\vec{x} - \vec{x}_i\|_2^3} \right)^T \\ \frac{f}{c} \frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} \\ 1 \end{bmatrix}^T \Bigg|_{\vec{x} = \hat{\vec{x}}_k, \dot{\vec{x}} = \hat{\vec{x}}_k} \quad (21.15)$$

GNSS-Based Velocity Estimation

GNSS signal processing of the carrier phase measurement provides the Doppler shift of the carrier frequency for velocity information. For GNSS, it is typical to estimate position and velocity with the Doppler shift and code/carrier phase measurements as these are already available to GNSS receivers. The first and simplest method for adding velocity estimation to the GNSS receiver-processor is to first estimate the GNSS receiver position, $\hat{\vec{x}}$, and clock offset, $\hat{f}\delta t$. Then, one can estimate the receiver velocity and clock drift as the second parameter vector, i.e.

$$\vec{\beta} = \begin{bmatrix} \dot{\vec{x}} \\ f\delta t \end{bmatrix} \quad (21.16)$$

Then, by rearranging the GNSS pseudorange rate equation

$$\Delta f_i + \frac{f}{c} \frac{(\vec{x} - \vec{x}_i)^T}{\sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2 + (\hat{z} - z_i)^2}} \dot{\vec{x}}_i = \frac{f}{c} \frac{(\vec{x} - \vec{x}_i)^T}{\sqrt{(\hat{x} - x_i)^2 + (\hat{y} - y_i)^2 + (\hat{z} - z_i)^2}} \dot{\vec{x}} + f\dot{\delta t} + \epsilon \quad (21.17)$$

which allows one to form the linear observation equation with $m \geq 4$ satellites

$$\vec{y} = \begin{bmatrix} \Delta f_1 + \frac{f}{c} \frac{(\hat{x} - \vec{x}_1)^T}{\|\hat{x} - \vec{x}_m\|} \dot{\vec{x}}_1 \\ \vdots \\ \Delta f_m + \frac{f}{c} \frac{(\hat{x} - \vec{x}_m)^T}{\|\hat{x} - \vec{x}_m\|} \dot{\vec{x}}_m \end{bmatrix} = X\vec{\beta} = \begin{bmatrix} \frac{f}{c} \frac{(\hat{x} - \vec{x}_1)^T}{\|\hat{x} - \vec{x}_1\|} & 1 \\ \vdots & \vdots \\ \frac{f}{c} \frac{(\hat{x} - \vec{x}_m)^T}{\|\hat{x} - \vec{x}_m\|} & 1 \end{bmatrix} \begin{bmatrix} \dot{\vec{x}} \\ f\delta t \end{bmatrix} \quad (21.18)$$

where the optimal estimate is given by the OLS solution

$$\hat{\vec{\beta}} = (X^T X)^{-1} X^T \vec{y} \quad (21.19)$$

The second and more accurate method would be to simultaneously estimate the velocity and position as a single parameter vector, i.e.

$$\vec{\beta} = \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \\ c\delta t \\ f\dot{\delta t} \end{bmatrix} \quad (21.20)$$

Then, the combined pseudorange and pseudorange rate equation allows one to form a nonlinear observation equation with m satellites as

$$\vec{y} = \begin{bmatrix} \rho_1 \\ \vdots \\ \rho_m \\ \Delta f_1 \\ \vdots \\ \Delta f_m \end{bmatrix} = \mathbf{f}(\vec{x}, \hat{\vec{\beta}}) = \begin{bmatrix} \|\vec{x} - \vec{x}_1\|_2 + c\delta t \\ \vdots \\ \|\vec{x} - \vec{x}_m\|_2 + c\delta t \\ \frac{(\vec{x} - \vec{x}_1)^T}{\|\vec{x} - \vec{x}_1\|_2} (\dot{\vec{x}} - \dot{\vec{x}}_1) + f\dot{\delta t} \\ \vdots \\ \frac{f}{c} \frac{(\vec{x} - \vec{x}_m)^T}{\|\vec{x} - \vec{x}_m\|_2} (\dot{\vec{x}} - \dot{\vec{x}}_m) + f\dot{\delta t} \end{bmatrix} \quad (21.21)$$

To solve this problem using NLS with GNA, assume one has an initial estimate, $\hat{\beta}_0$. Then, one can iteratively improve $\hat{\beta}$ by $\vec{\Delta}_k$ for $k = 0, 1, \dots$ as

$$\hat{\beta}_{k+1} = \hat{\beta}_k + \vec{\Delta}_k \quad (21.22)$$

where

$$\vec{\Delta}_k = \left(\mathbf{J}^T \mathbf{J} \right)^{-1} \mathbf{J}^T \left(\vec{y} - \mathbf{f}(\vec{x}, \hat{\beta}_k) \right) \quad (21.23)$$

where the i^{th} row of $2m \times 7 \mathbf{J}$ can be written for $i = 1, \dots, m$ as

$$\mathbf{J}_i = \begin{bmatrix} \frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} \\ \vec{0}_{3 \times 1} \\ 1 \\ 0 \end{bmatrix}^T \quad \vec{x} = \hat{\vec{x}}_k \quad (21.24)$$

and for $i = m + 1, \dots, 2m$ as

$$\mathbf{J}_i = \begin{bmatrix} \frac{f}{c} \left(\frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} - (\vec{x} - \vec{x}_1)^T (\vec{x} - \vec{x}_i) \frac{(\vec{x} - \vec{x}_1)^T}{\|\vec{x} - \vec{x}_1\|_2^3} \right) \\ \frac{f}{c} \frac{(\vec{x} - \vec{x}_i)^T}{\|\vec{x} - \vec{x}_i\|_2} \\ 0 \\ 1 \end{bmatrix}^T \quad \vec{x} = \hat{\vec{x}}_k, \dot{\vec{x}} = \hat{\dot{\vec{x}}}_k \quad (21.25)$$

The third and most accurate method would be to use a state filter to estimate the GNSS receiver position, velocity, clock offset, and clock drift, as a function of time, i.e.

$$\vec{x} = \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \\ c\delta t \\ f\dot{\delta t} \end{bmatrix} \quad (21.26)$$

Here, to form the prediction step for the *a priori* state, one could use a known vehicle dynamics model with stochastic inputs. Alternatively, one could also use a vehicle-agnostic dynamics model such as a **random motion model** which will be considered more fully in later chapters on target tracking. An alternative to this would be to use another navigation sensor, e.g. an inertial measurement unit, to form the prediction step. To form the correction step, one would use the nonlinear measurement at each time step as shown in the NLS method above, but with the additional consideration of additive white noise would be considered for both the code phase and the Doppler measurements. The *de facto* standard for GNSS filtering is the EKF, although the UKF and PF have seen some use.

21.2 Inertial Navigation Systems

An **inertial navigation system (INS)** is a multi-sensor navigation system that uses an inertial measurement unit (IMU) and a model-based data fusion algorithm to continuously calculate the position, velocity, and

attitude of a vehicle, i.e. the vehicle's **navigation state**. As INS do not require external references for computing a vehicle's position, un-aided INS perform dead reckoning (DR). An INS uses the IMU measurements of angular rate and specific force at every time step. Then, after initialization or the storage of the previous navigation state, one performs following three recursive updates:

1. Attitude update based on integrating the three-axis angular rate measurement,
2. Transformation of the specific force from body-fixed frame coordinates to navigation frame coordinates;
3. Gravitational force model calculation based on estimated position;
4. Velocity update based on integrating the acceleration; and
5. Position update based on integrating the velocity.

In INS design, there exists an inherent tradeoff among accuracy, processing efficiency, and model complexity. Before the 2000s, accuracy of the INS alone as well as processing efficiency were extremely important, thus, there exists many highly complex models for the INS equations. However, since the 2000s, most INS are aided with other sensors which allows a reduction in the complexity, e.g. GNSS coupling.

The discussion of all possible representations is beyond the scope of this textbook and a simple INS implementation for automotive/consumer grade IMUs is presented in this section. However, first, some notation must be introduced. In vehicle reference frames, recall the following frame subscripts

- I : Earth-Centered Inertial (ECI)
- E : Earth-Centered Earth Fixed (ECEF)
- N : navigation frame
- B : body-fixed frame

Furthermore, one can define the angular velocity of a frame A relative to frame B and expressed in frame C as

$$\vec{\omega}_{C,B \leftarrow A} \quad (21.27)$$

Lastly, recall the **skew-symmetric matrix operator**, i.e. $[\vec{x}]_{\times}$, which forms the following matrix from a 3-dimensional vector \vec{x}

$$[\vec{x}]_{\times} = \begin{bmatrix} 0 & -x_3 & x_2 \\ x_3 & 0 & -x_1 \\ -x_2 & x_1 & 0 \end{bmatrix} \quad (21.28)$$

Navigation State Definition

The vehicle's attitude is defined as the body-to-navigation frame transformation sequence of 3 – 2 – 1 Euler angles, i.e. yaw (ψ), pitch (θ), and roll (ϕ), grouped as the attitude vector

$$\vec{\psi}_{N \leftarrow B} = \begin{bmatrix} \phi \\ \theta \\ \psi \end{bmatrix} \quad (21.29)$$

The vehicle's velocity is defined in the North-East-Down (NED) navigation frame, \vec{v}_N , and the vehicle's position, \vec{p}_E , is expressed in geodetic coordinates, i.e. latitude (ℓ), longitude (λ), and altitude (h). These 9 states are updated in the INS algorithm presented here. However, the INS update equations must also use the direction cosine matrix (DCM), $C_{N \leftarrow B}$, which is related to the 3 – 2 – 1 Euler angles by

$$C_{N \leftarrow B} = \begin{bmatrix} \cos \theta \cos \psi & \sin \phi \sin \theta \cos \psi - \cos \phi \sin \psi & \cos \phi \sin \theta \cos \psi + \sin \phi \sin \psi \\ \cos \theta \sin \psi & \sin \phi \sin \theta \sin \psi + \cos \phi \cos \psi & \cos \phi \sin \theta \sin \psi - \sin \phi \cos \psi \\ -\sin \theta & \sin \phi \cos \theta & \cos \phi \cos \theta \end{bmatrix} \quad (21.30)$$

It should be noted that other INS algorithms propagate the DCM or quaternions. However, such update equations require constraints relating the independent and dependent parameters.

Thus, the INS update equations can be formalized by the following differential equations of motion.

$$\begin{aligned} \dot{\vec{\psi}}_{N \leftarrow B} &= \frac{1}{\cos \theta} \begin{bmatrix} 1 & \sin \phi \sin \theta & \cos \phi \sin \theta \\ 0 & \cos \phi \cos \theta & -\sin \phi \cos \theta \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \vec{\omega}_{B,N \leftarrow B} \\ \dot{\vec{v}}_N &= C_{N \leftarrow B} \vec{f}_B - [2\vec{\omega}_{N,I \leftarrow E} + \vec{\omega}_{N,E \leftarrow N}] \times \vec{v}_N + \vec{g}_N \\ \dot{\vec{p}}_E &= \begin{bmatrix} \dot{\ell} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_N(\ell)+h} & 0 & 0 \\ 0 & \frac{1}{(R_E(\ell)+h)\cos \ell} & 0 \\ 0 & 0 & -1 \end{bmatrix} \vec{v}_N = T \vec{v}_N \end{aligned} \quad (21.31)$$

where the attitude can alternatively be updated through the DCM directly as

$$\dot{C}_{N \leftarrow B} = [\vec{\omega}_{N,I \leftarrow B} - \vec{\omega}_{N,I \leftarrow N}] \times C_{N \leftarrow B} = C_{N \leftarrow B} [\vec{\omega}_{B,I \leftarrow B} - \vec{\omega}_{B,I \leftarrow N}] \times \quad (21.32)$$

where \vec{f}_B is the specific force expressed in the body frame, \vec{g}_N is the gravity vector in the navigation frame, R_N is the **north-south radius of curvature**, also known as the **meridian radius of curvature**, of the reference ellipsoid, i.e.

$$R_N(\ell) = \frac{a(1-e^2)}{(1-e^2 \sin^2 \ell)^{1.5}} \quad (21.33)$$

and R_E is **east-west radius of curvature**, also known as the **prime vertical radius of curvature**, of the reference ellipsoid, i.e.

$$R_E(\ell) = \frac{a}{\sqrt{1-e^2 \sin^2 \ell}} \quad (21.34)$$

where a is the semi-major axis, also known as the equatorial radius, of the reference ellipsoid and e is the eccentricity of the reference ellipsoid. Typically, one uses the WGS84 which provides the following values

$$a = 6,378,137 \text{ m} \quad (21.35)$$

and

$$e = \sqrt{f(2-f)} \quad (21.36)$$

where f is the WGS84 flattening, i.e.

$$f = \frac{1}{298.257223563} \quad (21.37)$$

For the initial attitude, if the vehicle is static and then the initial pitch and roll can be computed from a leveling procedure and, if navigation grade IMUs are being used, the initial heading or yaw can be computed from a gyrocompassing procedure. **Leveling** assumes the IMU is static where the accelerometers will only measure the gravity vector which is known to point in the down direction of the navigation frame whose direction in the body frame can be related to the pitch and roll. **Gyrocompassing** assumes the IMU is static where gyroscopes will only measure the Earth's rotation. However, the gyroscope errors must be adequately small with little drift in order to sense the Earth's rotation rate for a significant period of time. Alternatively, the initialization of the Euler angles may come from another sensor, e.g. a magnetometer which are sometimes included within IMUs. For the initial velocity estimate, if the vehicle is static, the inertial velocity is approximately the velocity of the Earth's surface at the initial position. Alternatively, if the vehicle is in motion, the initial velocity must be supplied by another sensor. For the initial position, one can start navigation at a known location or compute a position estimate using another sensor.

INS Attitude Update

The Euler angles, $\vec{\psi}_{N \leftarrow B}$, are related to the angular velocity of the body-to-navigation frame, $\vec{\omega}_{B,N \leftarrow B}$, through the following differential equation

$$\dot{\vec{\psi}}_{N \leftarrow B} = A(\vec{\psi}_{N \leftarrow B}) \vec{\omega}_{B,N \leftarrow B} \quad (21.38)$$

where the linear, parameter-varying (LPV) state matrix can be expressed as

$$A(\vec{\psi}_{N \leftarrow B}) = \frac{1}{\cos \theta} \begin{bmatrix} 1 & \sin \phi \sin \theta & \cos \phi \sin \theta \\ 0 & \cos \phi \cos \theta & -\sin \phi \cos \theta \\ 0 & \sin \phi & \cos \phi \end{bmatrix} \quad (21.39)$$

From the IMU, the rate gyroscopes measure the angular velocity of the body-to-inertial frame, i.e $\vec{\omega}_{B,I \leftarrow B}$ which requires the following correction

$$\vec{\omega}_{B,N \leftarrow B} = \vec{\omega}_{B,I \leftarrow B} - \vec{\omega}_{B,I \leftarrow N} \quad (21.40)$$

where $\vec{\omega}_{B,I \leftarrow N}$ can be computed by

$$\vec{\omega}_{B,I \leftarrow N} = C_{B \leftarrow N} (\vec{\omega}_{N,I \leftarrow N}) \quad (21.41)$$

$$\vec{\omega}_{B,I \leftarrow N} = C_{B \leftarrow N} (\vec{\omega}_{N,I \leftarrow E} + \vec{\omega}_{N,E \leftarrow N}) \quad (21.42)$$

where $\vec{\omega}_{N,I \leftarrow E}$ is the **Earth rotation rate** expressed in the navigation frame and can be computed using the formula

$$\vec{\omega}_{N,I \leftarrow E} = 7.292115 \times 10^{-5} \begin{bmatrix} \cos \ell \\ 0 \\ -\sin \ell \end{bmatrix} \text{ rad/s} \quad (21.43)$$

and $\vec{\omega}_{N,E \leftarrow N}$ is the **transport rate** expressed in the navigation frame and can be computed using the formula

$$\vec{\omega}_{N,E \leftarrow N} = \begin{bmatrix} \frac{v_{N,y}}{R_E(\ell)+h} \\ -\frac{v_{N,x}}{R_N(\ell)+h} \\ -\frac{v_{N,y} \tan \ell}{R_E(\ell)+h} \end{bmatrix} \quad (21.44)$$

However, when using automotive or consumer grade gyroscopes, $\vec{\omega}_{B,I \leftarrow N}$ is much smaller than the gyroscope errors and it is reasonable to assume that $\vec{\omega}_{B,N \leftarrow B} \approx \vec{\omega}_{B,I \leftarrow B}$, i.e. one is navigating on a non-rotating, flat-Earth.

In addition, the differential equation must be integrated to update the Euler angles from the previous time step(s) to the current. For automotive or consumer grade gyroscopes, simple Euler integration, a.k.a. a first order Runge-Kutta integration, can be used.

$$\vec{\psi}_{N \leftarrow B,k} = \vec{\psi}_{N \leftarrow B,k-1} + \Delta t A(\vec{\psi}_{N \leftarrow B,k-1}) \vec{\omega}_{B,N \leftarrow B} \quad (21.45)$$

where Δt is the time increment between steps $k - 1$ and k . Note that higher order Runge-Kutta integrations are also used, but require state estimates from multiple past time steps to be saved in memory.

INS Velocity Update

The NED velocity in the navigation frame, \vec{v}_N is related to the specific force, \vec{f}_B , normal gravity \vec{g}_N , and the Coriolis forces through the following differential equation

$$\dot{\vec{v}}_N = \vec{f}_N + \vec{g}_N - [2\vec{\omega}_{N,I \leftarrow E} + \vec{\omega}_{N,E \leftarrow N}]_{\times} \vec{v}_N \quad (21.46)$$

From the IMU, the accelerometers measure the specific force in the body frame, \vec{f}_B , which must be translated to the navigation frame, i.e.

$$\vec{f}_N = C_{N \leftarrow B} \vec{f}_B \quad (21.47)$$

and the local gravity vector is a function of latitude and altitude.

For automotive/consumer grade INS, the following gravity is typically adequate:

$$\vec{g}_N = \begin{bmatrix} 0 \\ 0 \\ g(\ell, h) \end{bmatrix} \quad (21.48)$$

where the gravity can be given by the **Somigliana gravity model**

$$g(\ell, h) = g_0(\ell) \left(1 - (3.157042870579883 \times 10^{-7} - 2.102689650440234 \times 10^{-9} \sin^2 \ell) h + (7.374516772941995 \times 10^{-14}) h^2 \right) \quad (21.49)$$

where

$$g_0(\ell) = 9.7803253359 \frac{1 + 0.001931853 \sin^2 \ell}{\sqrt{1 - 0.00669438 \sin^2 \ell}} \quad (21.50)$$

For automotive/consumer grade IMUs, the specific force, gravity, and Coriolis forces can be assumed to be constant across the time step Δt , thus providing a forwards Euler integration from previous time step $k - 1$ to current time step k as

$$\vec{v}_{N,k} = \vec{v}_{N,k-1} + \Delta t \dot{\vec{v}}_{N,k-1} \quad (21.51)$$

It should be noted that in this integration, the transformation of the specific force is assumed to be integrated over Δt with a constant rotation matrix $C_{N \leftarrow B,k}$ which is an approximation that can be improved with better integration such as averaging the values for $C_{N \leftarrow B,k}$ from the previous time step(s) and the current time step. In addition, it is important to note that often the Coriolis forces can even be dropped for low grade IMUs due to their effect being much less than the process noise. This implies that the navigation is being done on a flat, non-rotating Earth.

INS Position Update

The LLA position in the ECEF frame is related to the velocity through the differential equation

$$\dot{\vec{p}}_E = \begin{bmatrix} \dot{\ell} \\ \dot{\lambda} \\ \dot{h} \end{bmatrix} = \begin{bmatrix} \frac{1}{R_N+h} & 0 & 0 \\ 0 & \frac{1}{(R_E+h)\cos\ell} & 0 \\ 0 & 0 & -1 \end{bmatrix} \vec{v}_N \quad (21.52)$$

and can be integrated using a backwards Euler integration as

$$\vec{p}_{E,k} = \begin{bmatrix} \ell_k \\ \lambda_k \\ h_k \end{bmatrix} = \begin{bmatrix} \ell_{k-1} \\ \lambda_{k-1} \\ h_{k-1} \end{bmatrix} + \Delta t \begin{bmatrix} \frac{1}{R_N(\ell_{k-1})+h_{k-1}} & 0 & 0 \\ 0 & \frac{1}{(R_E(\ell_{k-1})+h_{k-1})\cos\ell_{k-1}} & 0 \\ 0 & 0 & -1 \end{bmatrix} \vec{v}_{N,k} \quad (21.53)$$

which assumes a constant integration period over Δt . It is also important to note that care should be taken in the precision of the values of ℓ and λ as 1 m of north-south displacement is approximately equivalent to 1.6×10^{-7} radians.

INS Errors and Performance

Another consideration in the INS equations is the propagation of error, not only in the equation models, but also in the angular velocity and specific force measurements. By performing a perturbation analysis of this model using a first-order Taylor series expansion, i.e. linearization, one can identify how the errors, represented by δ 's, propagate through the INS update equations, e.g.

$$\begin{aligned} \delta \dot{\vec{\psi}}_{N \leftarrow B} &\approx -[\vec{\omega}_{N,I \leftarrow N}]_x \delta \vec{\psi}_{N \leftarrow B} + \delta \vec{\omega}_{N,I \leftarrow N} - C_{N \leftarrow B} \delta \vec{\omega}_{B,I \leftarrow B} \\ \delta \vec{v}_N &= [C_{N \leftarrow B} \vec{f}_B]_x \delta \vec{\psi}_{N \leftarrow B} + C_{N \leftarrow B} \delta \vec{f}_B \\ &\quad - [2\vec{\omega}_{N,I \leftarrow E} + \vec{\omega}_{N,E \leftarrow N}]_x \delta \vec{v}_N \\ &\quad - [2\delta \vec{\omega}_{N,I \leftarrow E} + \delta \vec{\omega}_{N,E \leftarrow N}]_x \vec{v}_N + \delta \vec{g}_N \\ \delta \dot{\vec{p}}_E &= T' \delta \vec{p}_N + T \delta \vec{v}_N \end{aligned} \quad (21.54)$$

where the matrix T' relates the position errors to their time derivatives and $\delta \dot{\vec{\psi}}_{N \leftarrow B}$ is not the Euler angle errors, but the attitude errors resolved about the navigation frame. The relation of these errors to Euler angle errors will be discussed later. It should be noted that $\delta \vec{g}_N$ varies as a function of location, thus it is dependent on position errors which, in turn, lead to velocity errors. In particular, errors in altitude are significant as they lead to the well-known **vertical channel instability** problem. Thus, one typically uses some exteroceptive altimeter to form an aided INS.

Thus, INS performance is ultimately determined by the accelerometer and gyroscope measurement errors in the IMU. For these inertial sensors, simple errors models in the body frame can be chosen as

$$\delta \vec{f}_B = \vec{f}_B - (I_{3 \times 3} + M_a) \vec{f}_B + \vec{b}_a + \vec{w}_a \quad (21.55)$$

$$\delta \vec{\omega}_{B,I \leftarrow B} = \vec{\omega}_{B,I \leftarrow B} - (I_{3 \times 3} + M_g) \vec{\omega}_{B,I \leftarrow B} + \vec{b}_g + \vec{w}_g \quad (21.56)$$

where the a subscript represents the accelerometer parameters and g represents the gyroscope parameters. Here, \bar{f}_B and $\bar{\omega}_{B,I \leftarrow B}$ are the *true* specific force and angular velocity, M_a and M_g account for scale factor and axes misalignment errors, \vec{b}_a and \vec{b}_g are the bias vectors, and \vec{w}_a and \vec{w}_g are uncorrelated additive measurement noise.

Each of the error terms can be modeled in the INS algorithms as constant, time-varying, or a combination of both whose magnitude and variability of these errors depends on the IMU grade. These errors limit the performance of an INS. In particular, the accelerometer errors are integrated twice in the INS equations and lead to position errors that grow as a function of t^2 . likewise, gyroscope errors are integrated thrice and lead to position errors that grow as a function of t^3 . In order to mitigate the unbounded INS error growth, updates from exteroceptive sensor(s) are required which can be used to estimate these accelerometer and gyroscope biases forming an aided INS.

Notably, an AHRS-based DR solution only integrates the accelerometer errors once while a standard INS integrates the accelerometer errors twice. Thus, an ADHRS may be advantageous over a traditional INS solution if the airspeed errors are relatively less than the accelerometer errors.

21.3 GNSS-Aided Navigation Systems

GNSS and INS are complementary navigation systems utilizing three strengths of the INS and one strength of GNSS. First, the absence of an external signals for INS makes it more reliable and secure than GNSS. Second, the INS estimation rate is only limited by the processing power of the onboard computer which can be as fast as 100 Hz or more while most GNSS receivers only update from 1-20 Hz. Third, GNSS provides measurements for inferring position and velocity, but not attitude, while INS must estimate the attitude. However, one should note that multiple GNSS receivers with a known baseline can be used to estimate attitude. Fourth, INS output errors are time-correlated and unbounded (when using low-cost inertial sensors) while a GNSS receiver has bounded errors on its position and velocity estimates.

Thus, integration of GNSS and INS provides estimates to arrest the effect of the INS errors through the GNSS measurements, while the INS provides high bandwidth attitude, position, and velocity for vehicle guidance and control. There are many different integration schemes for the data fusion of GNSS and INS, but in general there are three different types: loose, tight, and deep. Though some crossover and ambiguity may exist between the three, these distinctions will assist in explaining the different concepts in GNSS/INS integration.

A diagram of a typical **loose GNSS/INS integration**, a.k.a. loosely-coupled, also known as **position-domain integration**. Here, the INS and GNSS algorithms operate as independent navigation systems and the navigation solution from each is blended using an estimator to form a third navigation solution. The primary variant to this scheme is the inclusion or exclusion of the dashed-line as a feedback signal in the blending software leading to either a closed-loop or open-loop configuration. The inclusion of the feedback signal is necessary for low-grade INS, but for navigation grade, this is not necessary, but does cause the INS to be dependent on the GNSS. The primary strength to this approach is the independence of solutions, thus faults in either do not affect the other. Another advantage is the simple implementation, though time synchronization can still be a challenge. When a consumer grade INS is used, then the INS simply supplies attitude information and increased bandwidth to the navigation system.

A diagram of a typical **tight GNSS/INS integration** a.k.a. tightly-coupled, also known as **range-domain integration**. This design blends the sensor measurements together instead of the navigation solution. The

primary variant to this scheme is the inclusion or exclusion of **tracking loop aiding** which feeds the blending filter's position and/or velocity estimate back to the GNSS receiver to enhance its performance. This is often necessary for aircraft in high-dynamic maneuvers to be able to continuously track the GNSS signals. However, this integration adds complexity to the GNSS tracking loops and creates a dependency between the systems which does not allow for separate faults to be isolated. The primary advantage is that tight integration provides a more accurate and robust solution than loose integration. The enhanced accuracy is, in part, due to the fact that the GNSS measurements exhibit less temporal correlation than the INS navigation solution. It is robust because it can use the GNSS measurements when there are less than four satellites, though this will only be possible for a limited amount of time due to GNSS receiver clock drift.

A diagram of a typical **deep GNSS/INS integration** a.k.a. deeply-coupled, also known as **tracking-domain integration**. The primary variant to this scheme is coherent or noncoherent. This scheme is an optimal fusion of the INS and GNSS measurements as it utilizes the IMU measurements in a **vector delay lock loop (VDLL)**, enhancing the robustness of the GNSS receiver to interference and jamming even better than tight integration. However, this design introduces enhanced complexity to integrating the GNSS and INS with the necessity of access to the receiver hardware or a software-defined receiver (SDR) for the signal processing.

As the 3-DOF attitude dynamics in navigation are highly nonlinear integrations of Euler angles, DCMs, or quaternions, one often uses **error-state filtering** as the multi-sensor data fusion algorithm. For GNSS/INS integration, error-state filtering uses a time update step for the INS and measurement update step for estimating the error-state of the INS using the GNSS measurements. Then, the additional state update step corrects the nominal-state INS estimate with the GNSS-based error-state estimate. In navigation, the *de facto* standard algorithm for data fusion is the error-state EKF although the UKF and the PF have been used in some cases. This section will describe the three update steps for and ES-EKF implementation of a loose GNSS/INS integration system.

ES-EKF for Loose GNSS/INS Integration

For the GNSS/INS integration, the error-state dynamics model can vary greatly based on IMU grade, the GNSS update accuracy and frequency, and the overall vehicle mission. For simplicity, the ES-EKF here will only use the accelerometer and gyroscope biases which assumes one has sufficiently calibrated any axes misalignment and scale factor errors. Thus, the loose GNSS/INS integration nominal-state estimated by the INS is

$$\bar{x} = [\bar{p}^T \quad \bar{v}_N^T \quad \bar{\psi}_{N \leftarrow B}^T \quad \bar{b}_a^T \quad \bar{b}_g^T]^T \quad (21.57)$$

where $\bar{p} = [\bar{\ell} \quad \bar{\lambda} \quad \bar{h}]^T$ is the LLA position vector, \bar{v}_N is the NED velocity vector, $\bar{\psi}_{N \leftarrow B} = [\bar{\phi} \quad \bar{\theta} \quad \bar{\psi}]^T$ is the Euler angle vector, \bar{b}_a is the accelerometer bias vector, and \bar{b}_g is the gyroscope bias vector. Likewise, the loose GNSS/INS integration error-state vector is

$$\delta \vec{x} = [\delta \bar{p}^T \quad \delta \bar{v}_N^T \quad \delta \bar{\psi}_{N \leftarrow B}^T \quad \delta \bar{b}_a^T \quad \delta \bar{b}_g^T]^T \quad (21.58)$$

where $\delta \bar{p} = [\delta x \quad \delta y \quad \delta z]^T$ is the NED position error-state, $\delta \bar{v}_N$ is the NED velocity error-state, $\delta \bar{\psi}_{N \leftarrow B}$ is the Euler angle error-state, $\delta \bar{b}_a$ is the accelerometer bias error-state, and $\delta \bar{b}_g$ is the gyroscope bias error. Notably, in a tight GNSS/INS integration utilizing pseudorange and pseudorange rate, one would need to include the clock offset and clock drift error-states, δt and $\dot{\delta t}$.

For the ES-EKF state update step, one must update the nominal-state INS estimates. For the LLA position update, one has

$$\hat{\vec{p}} = \begin{bmatrix} \hat{\ell} \\ \hat{\lambda} \\ \hat{h} \end{bmatrix} = \begin{bmatrix} \hat{\ell} \\ \hat{\lambda} \\ \hat{h} \end{bmatrix} + \begin{bmatrix} \frac{\delta x}{R_N(\hat{\ell})+\hat{h}} \\ \frac{\delta y}{(R_E(\hat{\ell})+\hat{h}) \cos \hat{\ell}} \\ -\delta z \end{bmatrix} \quad (21.59)$$

where $R_N(\hat{\ell})$ is the **north-south radius of curvature** and $R_E(\hat{\ell})$ is **east-west radius of curvature** of the reference ellipsoid. For the NED velocity update, one has

$$\hat{\vec{v}} = \hat{\vec{v}} + \delta \hat{\vec{v}} \quad (21.60)$$

However, for low-grade IMUs, often one relies only on GNSS-based estimates of \hat{h} and \hat{z} updates, due to the vertical channel instability problem.

For the accelerometer bias update, one has

$$\hat{\vec{b}}_a = \hat{\vec{b}}_a + \delta \hat{\vec{b}}_a \quad (21.61)$$

and for the gyroscope bias update, one has

$$\hat{\vec{b}}_g = \hat{\vec{b}}_g + \delta \hat{\vec{b}}_g \quad (21.62)$$

It should be noted that the accelerometer and gyroscope bias estimates are added to the raw INS measurements before use in the INS equations.

For the Euler angle attitude update, one must first correct the DCM using the error-state estimate. Using a first-order approximation for the DCM error, one has

$$C_{N \leftarrow B} = \left(I_{3 \times 3} - \left[\delta \hat{\vec{\psi}}_{N \leftarrow B} \right]_{\times} \right) C_{N \leftarrow B} \quad (21.63)$$

Then, the updated Euler angles can be extracted from $C_{N \leftarrow B}$ by

$$\hat{\vec{\psi}}_{N \leftarrow B} = \begin{bmatrix} \hat{\phi} \\ \hat{\theta} \\ \hat{\psi} \end{bmatrix} = \begin{bmatrix} \tan^{-1} \frac{C_{32}}{C_{33}} \\ -\sin^{-1} C_{31} \\ \tan^{-1} \frac{C_{21}}{C_{11}} \end{bmatrix} \quad (21.64)$$

where C_{ij} is the element in the i^{th} row and j^{th} column of $C_{N \leftarrow B}$.

For the ES-EKF time update step, the accelerometer's specific force measurement *error* can be modeled as

$$\delta \vec{f}_B = \delta \vec{b}_a + \vec{w}_a \quad (21.65)$$

and the gyroscope's angular rate measurement *error* can be modeled as

$$\delta \vec{\omega}_{B,I \leftarrow B} = \delta \vec{b}_g + \vec{w}_g \quad (21.66)$$

where \vec{w}_a and \vec{w}_g are additive zero-mean IMU noise processes with uncorrelated power spectral densities (PSD) of $\sigma_{w_a}^2$ and $\sigma_{w_g}^2$, respectively. Notably, these IMU noise processes don't affect the biases, but do affect the Euler angle, velocity, and position estimates due to the INS integrations of the IMU measurements.

For low-grade IMUs, one typically assumes these biases evolve in time according to first-order Gauss-Markov processes, i.e.

$$\dot{\delta \vec{b}_a} = -\frac{1}{\tau_a} \delta \vec{b}_a + I_{3 \times 3} \vec{\mu}_a \quad (21.67)$$

and

$$\dot{\delta \vec{b}_g} = -\frac{1}{\tau_g} \delta \vec{b}_g + I_{3 \times 3} \vec{\mu}_g \quad (21.68)$$

where μ_a and μ_g are the zero-mean correlated Markov bias errors with PSDs as

$$\sigma_{\mu_a}^2 = \frac{2\sigma_{b_a}^2}{\tau_a} \quad (21.69)$$

and

$$\sigma_{\mu_g}^2 = \frac{2\sigma_{b_g}^2}{\tau_g} \quad (21.70)$$

Thus, the stacked IMU noise vector contains the IMU output noise and first-order Markov noise as

$$\vec{w} = \begin{bmatrix} \vec{w}_a \\ \vec{w}_g \\ \vec{\mu}_a \\ \vec{\mu}_g \end{bmatrix} \quad (21.71)$$

with PSD

$$S_\omega = \begin{bmatrix} \sigma_{w_a}^2 I_{3 \times 3} & 0 & 0 & 0 \\ 0 & \sigma_{w_g}^2 I_{3 \times 3} & 0 & 0 \\ 0 & 0 & \sigma_{\mu_a}^2 I_{3 \times 3} & 0 \\ 0 & 0 & 0 & \sigma_{\mu_g}^2 I_{3 \times 3} \end{bmatrix} \quad (21.72)$$

Taken together, the linearized error-state dynamics equation is given by

$$\dot{\delta \vec{x}}(t) = A(t) \delta \vec{x}(t) + L(t) \vec{w}(t) \quad (21.73)$$

where the time-varying state matrix, A , is approximated from the linearized INS error equations and bias models as

$$A(t) = \begin{bmatrix} -[\vec{\omega}_{N,E \leftarrow N}]_x & I_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ \frac{\|\vec{g}\|_2}{a} \text{diag}(-1, -1, 2) & -[2\vec{\omega}_{N,I \leftarrow E} + \vec{\omega}_{N,E \leftarrow N}]_x & [\vec{C}_{N \leftarrow B} \vec{f}_B]_x & \vec{C}_{N \leftarrow B} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & -[\vec{\omega}_{N,I \leftarrow N}]_x & 0_{3 \times 3} & -\vec{C}_{N \leftarrow B} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & -\frac{1}{\tau_a} I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & -\frac{1}{\tau_g} I_{3 \times 3} \end{bmatrix} \quad (21.74)$$

where $\|\vec{g}_N\|$ is the local gravity magnitude and depends on latitude and altitude and a is the semi-major axis of the Earth, a.k.a. equatorial radius. Furthermore, the time-varying process noise gain matrix L is

approximated from the INS noise models transformed to the correct frame as

$$L(t) = \begin{bmatrix} 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ C_{N \leftarrow B} & 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & -C_{N \leftarrow B} & 0_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} & 0_{3 \times 3} \\ 0_{3 \times 3} & 0_{3 \times 3} & 0_{3 \times 3} & I_{3 \times 3} \end{bmatrix} \quad (21.75)$$

For the ES-EKF measurement update step, one must compute the innovation between the expected measurement based on the INS navigation state and the actual GNSS measurements, i.e.

$$\delta \vec{y}_k = \vec{y}_k - \hat{\vec{y}}_k \quad (21.76)$$

For the loose GNSS/INS integration ES-EKF, the GNSS measurements are the position vector, \vec{p} , and velocity, \vec{v} , which are typically in the LLA frame and the NED frame, respectively. However, as the position error is in NED as that typically is how the measurement error covariance is defined, one is required to transform the nominal-state and GNSS position measurements to the *same* NED navigation frame with the INS nominal-state as the $[0 \ 0 \ 0]^T$ reference point of the NED frame. Then, the expected INS measurement model can relate the GNSS measurement to the nominal-state estimate in NED coordinates as

$$\hat{\vec{y}}_k = H_k \hat{\vec{x}}_k \quad (21.77)$$

and the linearized GNSS measurement model is

$$\vec{y}_k = H_k \vec{x}_k \quad (21.78)$$

where

$$H_k = [I_{6 \times 6} \ 0_{6 \times 9}] \quad (21.79)$$

The GNSS measurement noise covariance is typically modeled as

$$R_k = \begin{bmatrix} P_{p,k} & 0 \\ 0 & P_{v,k} \end{bmatrix} \quad (21.80)$$

where $P_{p,k}$ and $P_{v,k}$ are the estimated position and velocity covariance matrix from the GNSS state estimator, respectively. However, an important consideration is that GNSS noise is truly auto-correlated, thus violating the Markov process assumptions for the this Bayes filtering method. Thus, in theory, one should model and estimate this auto-correlation in the filter. However, in practice, one inflates the covariance matrix to compensate for this. This is sub-optimal, but with low-grade INS, one does not necessarily desire to drastically improve the GNSS positioning, but to simply directly estimate attitude and have a higher bandwidth for the vehicle state estimate provided by the overall navigation system.

GNSS/INS Integration Implementation Considerations

When implementing a GNSS/INS integrated system, an important consideration is the spatial offset between the IMU and the GNSS antenna which should be compensated for when fusing the information from the two sensors together since they are referring to two different points of the vehicle. This spatial offset is typically

called the **lever arm**, parameterized in the body frame, and denoted by \vec{l}_B . The position and velocity of the GNSS and INS in the navigation frame can be related by the translation equations

$$\vec{p}_{N,GNSS} = \vec{p}_{N,INS} + C_{N \leftarrow B} \vec{l}_B \quad (21.81)$$

and

$$\vec{v}_{N,GNSS} = \vec{v}_{N,INS} + C_{N \leftarrow B} \left([\vec{\omega}_{B,I \leftarrow B}]_\times + [\vec{\omega}_{B,I \leftarrow N}]_\times \right) \vec{l}_B \quad (21.82)$$

where the accuracy of the translation depends on the accuracy of the lever arm as well as the gyroscope accuracy for the velocity. In a loose GNSS/INS integration, one typically translates the GNSS position and velocity to the IMU before the error measurement update step, while for a tight GNSS/INS integration the INS position and velocity are typically translated to the GNSS antenna before fusing the error measurement update step.

A second related aspect of fusing two different sensors is **time synchronization** which must be handled by the dedicated hardware or data collection systems whose design is beyond the scope of this textbook. However, one should be aware that a timing error δt will affect the velocity and position according to Newtonian physics in the presence of an acceleration, e.g.

$$\delta v = a\delta t \quad (21.83)$$

$$\delta p = \frac{1}{2}a\delta t^2 \quad (21.84)$$

Thus, a timing error of 1 ms and an acceleration of 10 m/s², the position and velocity error would be $\ll 1$ mm and 1 cm/s, respectively, which shows that the position error is negligible, though the velocity error may be problematic for higher precision navigation systems. Lastly, when implementing the covariance updates in an EKF, one should force the covariance to be symmetric.

21.4 Baro- and Terrain-Aided Navigation Systems

Baro-Aided Navigation Systems

Terrain-Aided Navigation Systems

Similar to GNSS, another common global solution for aiding INS with positioning is to use a terrain database that can be continuously detected, identified, and tracked by a radar or electro-optical sensor, e.g., lidar or camera. Here, the continuous terrain or the features can be used to form pseudorange measurements for positioning that aid the INS. This section will discuss the particulars of using terrain databases to aid dead reckoning and inertial navigation systems.

Terrain-relative navigation (TRN), also known as **terrain-referenced navigation (TRN)**, typically consists of

21.5 Feature Map-Aided Navigation Systems

Here if the features **simultaneous localization and mapping (SLAM)**

Introductory Target Tracking Systems

22.1 Introduction to Target Tracking Systems

Target tracking is similar to navigation, in that one desires to estimate a vehicle's state. However, **navigation** is the state estimation of own vehicle's position, velocity, attitude, i.e. the **navigation state** vector while **target tracking** is the state estimation of another vehicle's position, velocity, perhaps attitude, i.e. the **target state** vector. Thus, similar energy wave sensing technologies used for navigation are also used for target tracking. This is a subset of **object tracking** which can be divided into static and dynamic objects as well as passive dynamic or actively controlled objects. Recall that static and passive dynamic objects are also known as **feature** and tracking these features is known as **feature tracking** which is easier than target tracking.

The sensors in target tracking systems use signal processing or image processing to derive measurements from the electrical signals converted from the energy waves that provide vehicle state information through physical phenomena, also known as **pseudomeasurements**. The general types of pseudomeasurements are **pseudoranges**, **pseudoazimuths**, **pseudoelevations**, and **pseudorange rates**. The processing details for these various target tracking sensors will only be introduced in this section and the theory behind their design is beyond the scope of this textbook.

There are two fundamental challenges in target tracking that drive the different approaches in tracking system design. The first challenge is that the tracking system must first detect the target(s) to be tracked. Thus, the target detection algorithms will exhibit missed detections and false alarms, known as **clutter** in target tracking, also known as the **measurement origin uncertainty**. Furthermore, one may not know the number of targets to track which may dynamically increase, known as **target birth** and **target spawning** in target tracking, or decrease, also known as **target death** in target tracking. This challenge must be accounted for in target tracking algorithms and lead to the different approaches. This introductory chapter of target tracking will consider target tracking methods when the number of targets is known and constant. The following advanced chapter will consider target tracking methods for an unknown and time-varying number of targets.

Clutter refers to extracted measurements from the images that are returned from "targets" which are uninteresting to the tracking system. Such targets include natural objects such as ground, sea, precipitation,

sand storms, birds, atmospheric turbulence, ionosphere reflections, meteor trails, and buildings. Clutter is considered a passive interference source, since it only appears in response to radar signals sent by the radar. Clutter is detected and neutralized in several ways. Clutter tends to appear static between images; on subsequent scan echoes, desirable targets will appear to move, and all stationary echoes can be eliminated. Clutter may also originate from multipath echoes from valid targets caused by ground reflection, atmospheric ducting or ionospheric reflection/refraction. This clutter type is especially bothersome since it appears to move and behave like other normal point targets of interest. These problems can be overcome by incorporating a ground map of the radar's surroundings and eliminating all echoes which appear to originate below ground or above a certain height.

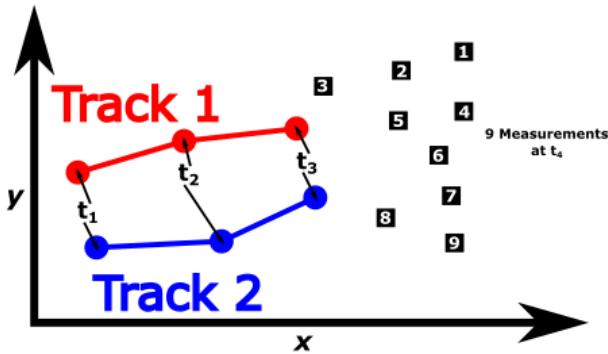
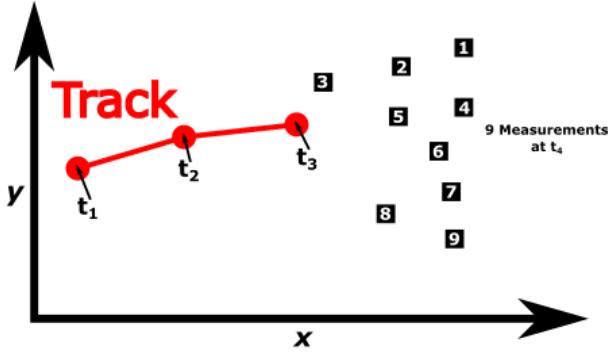
An extension of this challenge are the use of **superpositional sensors** where each target can contribute to any number of measurements, also known as **split measurements**; each measurement is potentially affected by multiple targets in an additive fashion, also known as **merged measurements**; and measurements are not independent. Other advanced topics beyond the scope of this textbook as current areas of active research include **multi-sensor, multi-target tracking** including distributed multi-target tracking with **sensor networks** of heterogeneous and geographically dispersed nodes with sensing, communication, and processing capabilities as well as **multi-tracker fusion** where one desires to fuse the estimated targets from multiple tracking systems.

The second challenge is that the tracking system can only use remote sensing to estimate target state. Thus, the control inputs of the target vehicle and/or inertial sensors cannot be used as part of the time update in the traditional Bayes filtering approach used in navigation, also known as the **target motion uncertainty**. Thus, target tracking depends on general motion models for the targets one desires to track. Thus, this chapter of the textbook will discuss motion models for **point(-mass) target tracking**, i.e. the target state contains dynamic characteristics and not spatial characteristics and only single detections per target are possible. This is opposed to **extended(-mass) target tracking** where the target state contains spatial characteristics or **group target tracking** where multiple targets move according to some common group dynamics. In general, point target motion models can be distinguished between the amount of coupling assumed between the coordinates of the vehicle's motion as will be discussed in this introductory chapter.

Measurement Origin Uncertainty

The optimal state estimator is the Bayes filter which for linear, Gaussian stochastic systems is the Kalman filter. However, as any target detection procedure will exhibit absent and cluttered measurements of the N target(s), target tracking must account for target-originated measurements and clutter-originated measurements, i.e. the **measurement origin uncertainty**, also known as the **data association problem**, when performing the **measurement update step** or **correction step** within the Bayes filter context.

This measurement origin uncertainty can be demonstrated using the following graphics. Here, at some time $k = t_4$, one is tracking $N = 1$ or 2 targets and receives $M = 9$ measurements, i.e. **sensor data**, which contain a combination of clutter and possibly the target-originated measurements.



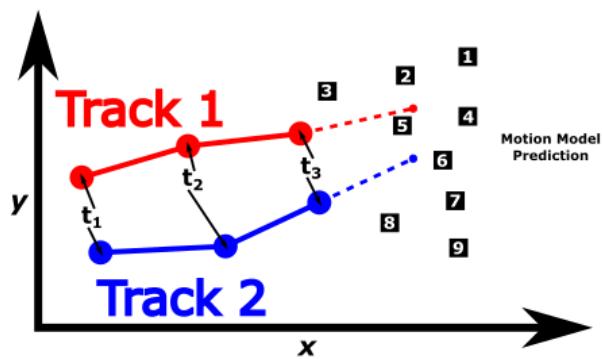
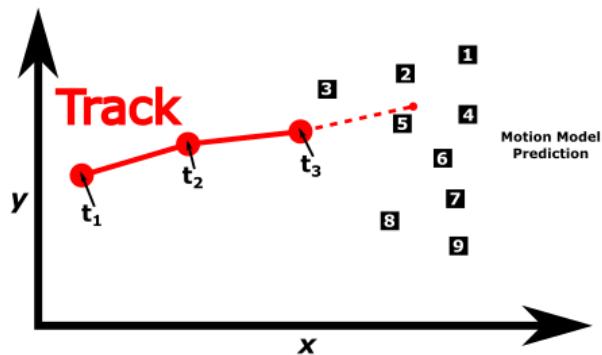
In the Bayes filter context, one can perform the prediction step of the target state estimator, i.e. obtain some $\hat{\vec{x}}_{k|k-1}$ by using a chosen process equation

$$\hat{\vec{x}}_{k|k-1} = f \left(\hat{\vec{x}}_{k-1|k-1}, \hat{\vec{w}}_{k-1} \right) \quad (22.1)$$

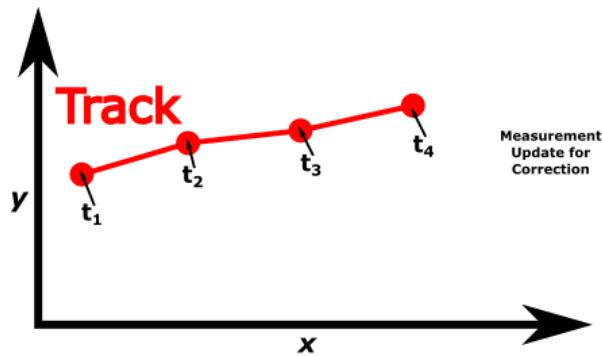
which could be any of the target motion models discussed in previous sections. This will notably also correspond to some predicted measurement, $\hat{\vec{y}}_{k|k-1}$, through the measurement equation, i.e.

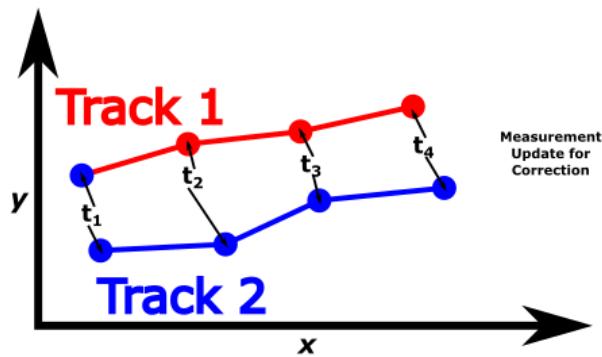
$$\hat{\vec{y}}_{k|k-1} = h \left(\hat{\vec{x}}_{k|k-1}, \hat{\vec{v}}_k \right) \quad (22.2)$$

which could be any of the pseudomeasurements for target tracking. This step can be represented by the small circle in the following figure.



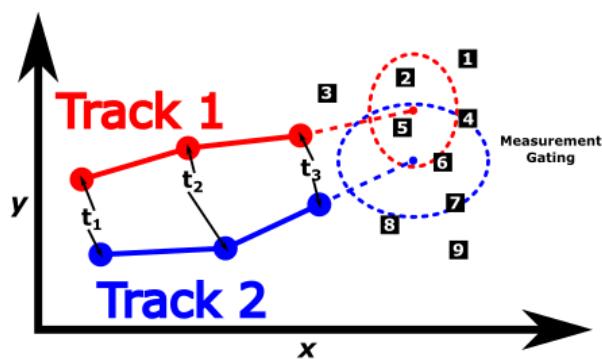
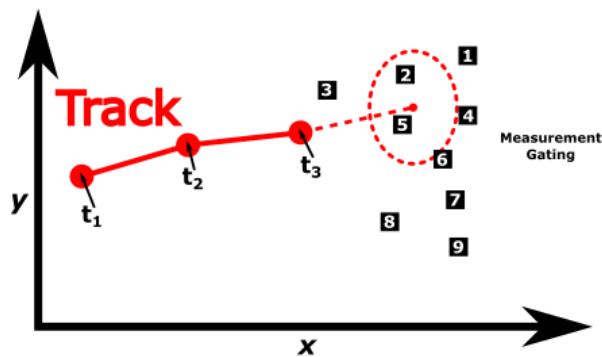
At this point, one must perform some **track-to-measurement association**, also known as **track-to-sensor data association** in order to generate a Bayes-like **correction step**, also known as the **measurement update**, on the predicted target state, i.e. obtain some $\hat{x}_{k|k}$ regardless of the choice of data association method. This can be represented by the big circle in the following figure.





Measurement Gating

Theoretically, to account for all possible measurements as target-originated, one should consider all possible associations between the predicted track and the sensor data, i.e. the **candidate measurement set**. However, if significant clutter is present, one often performs a **gating step** which reduces the candidate measurement set to contain only those within some partition of the measurement space, \mathbb{R}^{n_y} before performing the data association. This partitioning procedure typically depends on using some statistical characteristics relating the measurements to the predicted measurement. This concept is represented in the following figures.



where the gate is the ellipse centered on the predicted state estimate. The inside of the ellipse is the partition of measurements which should be considered as possibly target-originating, i.e. some M' subset of the M measurements $\vec{y}_k \in \mathbb{R}^{n_y}$.

When designing the gate, one typically specifies the **gate coverage**, $1 - \alpha$, defined as the probability of including a target-originated measurement, i.e.

$$\Pr(\vec{y}_k \in \mathcal{G}) \geq 1 - \alpha \quad (22.3)$$

Then, assuming that the innovation is Gaussian distributed, i.e. $\vec{y}_k - \hat{\vec{y}}_{k|k-1} \sim \mathcal{N}(0, S)$, two common gates used are rectangular gates and ellipsoidal gates.

Rectangular gates (RG) partition measurements inside a hyper-rectangular region, i.e.

$$\mathcal{G}_R = \left\{ \vec{y}_k \in \mathbb{R}^{n_y} : |\vec{y}_k - \hat{\vec{y}}_{k|k-1}|_j \leq F_N^{-1} \left(1 - \frac{\alpha}{n_y} \right) S_{jj}^{0.5}, \quad j = 1, \dots, n_y \right\} \quad (22.4)$$

where $|\bullet|_j$ is the j^{th} element of $|\bullet|$, F_N is the standard normal distribution and $F_N^{-1} \left(1 - \frac{\alpha}{n_y} \right) S_{jj}^{0.5}$ is the **rectangular gating threshold**. Then, the gate coverage can be shown to be bounded by noting that

$$\Pr(\vec{y}_k \in \mathcal{G}_R) = \Pr \left(\bigcup_{j=1}^{n_y} \left(|\vec{y}_k - \hat{\vec{y}}_{k|k-1}|_j \leq F_N^{-1} \left(1 - \frac{\alpha}{n_y} \right) S_{jj}^{0.5} \right) \right) \quad (22.5)$$

which by Bonferroni's inequality, one has

$$\Pr(\vec{y}_k \in \mathcal{G}_R) \geq 1 - \sum_{j=1}^{n_y} \Pr \left(|\vec{y}_k - \hat{\vec{y}}_{k|k-1}|_j > F_N^{-1} \left(1 - \frac{\alpha}{n_y} \right) S_{jj}^{0.5} \right) \quad (22.6)$$

and simplifying, one has

$$\Pr(\vec{y}_k \in \mathcal{G}_R) \geq 1 - \frac{n_y \alpha}{n_y} \quad (22.7)$$

or

$$\Pr(\vec{y}_k \in \mathcal{G}_R) \geq 1 - \alpha \quad (22.8)$$

Ellipsoidal gates (EG) ignore any measurements outside the ellipsoidal region, i.e.

$$\mathcal{G}_E = \left\{ \vec{y}_k \in \mathbb{R}^{n_y} : (\vec{y}_k - \hat{\vec{y}}_{k|k-1})^T S^{-1} (\vec{y}_k - \hat{\vec{y}}_{k|k-1}) \leq F_{\chi(n_y)}^{-1} (1 - \alpha) \right\} \quad (22.9)$$

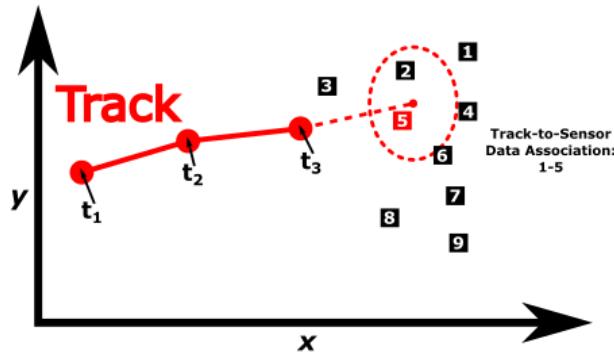
where $F_{\chi(n_y)}$ is the centralized χ^2 -distribution with n_y degrees-of-freedom and $F_{\chi(n_y)}^{-1} (1 - \alpha)$ is the **ellipsoidal gating threshold**. Furthermore, by the eigendecomposition of S as $V^T \Lambda V$ and defining $\vec{y}_k - \hat{\vec{y}}_{k|k-1}$ as \vec{r}_k and $V \vec{r}_k$ as \tilde{r}_k , one has

$$\vec{r}_k^T V^T \Lambda^{-1} V \vec{r}_k = \tilde{r}_k^T \Lambda^{-1} \tilde{r}_k = \sum_{i=1}^{n_y} \frac{\text{num}}{\text{den}} \sim \chi^2(n_y) \quad (22.10)$$

which proves the gate coverage is exactly $1 - \alpha$. Thus, for a given α and a truly Gaussian distribution for the innovation, the ellipsoidal gate is optimal in the minimal **gate volume** sense and thus is used most commonly. However, in practical tracking systems, one typically enlarges this ellipsoidal gate based on possible target maneuvers that may not be captured by the motion model prediction, also known as a **two-stage gating step**. Other types of gates are also possible if additional data and/or simulations are available for the tracking system design.

Single-Target Data Association Methods

For single-target data association, the simplest method would be to use only the **nearest-neighbor (NN) association** which selects a *single* measurement with the smallest “distance” to the predicted measurement, i.e. $\hat{y}_{k|k-1}$. This concept is represented in the following figure.



Here, a statistical distance measure is typically used, either with the mean distance, i.e. the Euclidean distance

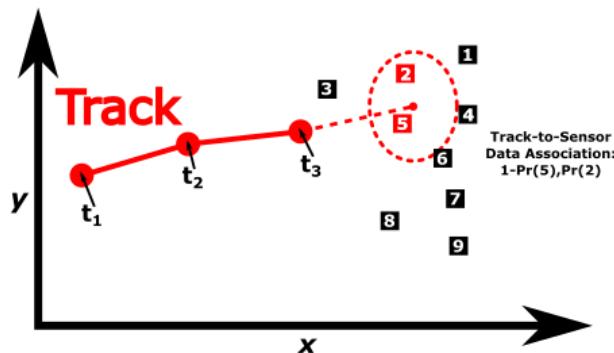
$$\vec{y}_k = \underset{\vec{y}_k \in \mathbb{R}^{n_y}}{\operatorname{argmin}} \|\vec{y}_k - \hat{y}_{k|k-1}\|_2^2 \quad (22.11)$$

or the covariance-weighted distance, i.e. the **Mahalanobis distance**

$$\vec{y}_k = \underset{\vec{y}_k \in \mathbb{R}^{n_y}}{\operatorname{argmin}} \left(\vec{y}_k - \hat{y}_{k|k-1} \right)^T S^{-1} \left(\vec{y}_k - \hat{y}_{k|k-1} \right) \quad (22.12)$$

In addition, one could compute additional “affinity scores” for each measurement based on expected characteristics, e.g. size, shape, appearance. Regardless, NN association allows one to use a standard Bayes filter correction step with the selected measurement.

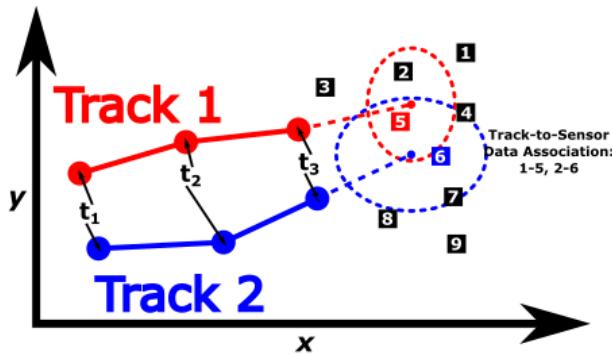
A second data association method would be to use the **all-neighbors (AN) association**, also known as the **probabilistic data association (PDA)**, which uses *all* measurements within the gate, but weights each probabilistically in the correction step of the resulting filter. This concept is represented in the following figure as



However, to weigh each measurement probabilistically, one must know have a statistical model for the probability of missed detections and clutter, as well as the process and measurement models.

Multi-Target Data Association Methods

For multi-target tracking, the simplest method is **global nearest-neighbor (GNN) association** which assigns a *single* measurement to *each* predicted measurement, i.e. associates a single set of $\hat{y}_{i,k|k-1}$ for $i = 1, \dots, N$ to a *single* $\vec{y}_{j,k|k-1}$ for $j = 1, \dots, M'$, based on the smallest sum of “distances”. This concept is represented in the following figure.

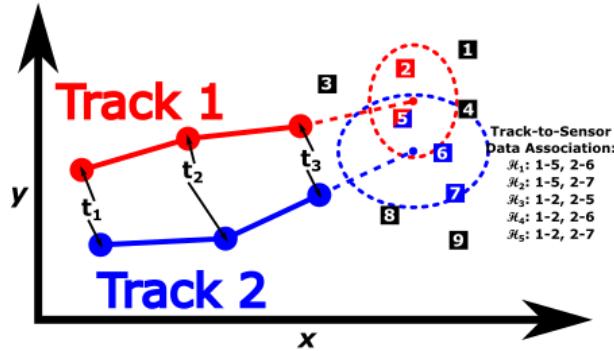


This **assignment problem** seeks to minimize the overall distance, i.e. cost, of each assignment of the N tracks to the gated M' measurements which can be efficiently solved in polynomial time using the **Hungarian algorithm** which is outlined as follows for $N < M'$.

1. Form the $M' \times M'$ cost matrix:
 - ij element for $i < N$: distance between i^{th} track and j^{th} measurement
 - Pad with small random numbers for $M' - N \times M'$ sub-matrix
2. Subtract the minimal cost from each row
 - If one can form a permutation matrix out of 0 elements in $N \times M'$ submatrix, stop
3. Subtract the minimal cost from each column
 - If one can form a permutation matrix out of 0 elements in $N \times M'$ submatrix, stop
4. Draw as few row or column lines as possible to cover all 0's
5. From the elements remaining, subtract lowest value from all elements not struck and add to elements at the intersection of row/column lines
 - If one can form a permutation matrix out of 0 elements in $N \times M'$ submatrix, stop
 - If not go back to step four until finished

A second, closely related, data association method would be to use the k **nearest-neighbors (k -NN) association**, which uses k combinations of measurement to *each* predicted measurement, i.e. associates k

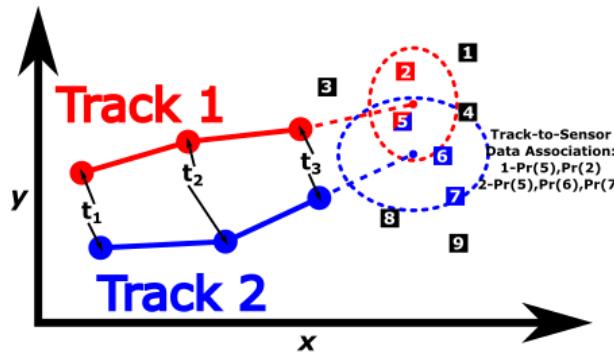
sets of $\hat{y}_{i,k|k-1}$ for $i = 1, \dots, N$ to the set of $\vec{y}_{j,k|k-1}$ for $j = 1, \dots, M'$, based on the k smallest sums of “distances”. This concept is represented in the following figure.



This **k -best assignment problem** seeks to minimize the overall distance, i.e. cost, of each assignment of the N tracks to the M' measurements which can be efficiently solved in polynomial time using the **Murty's algorithm** which is outlined as follows for $N < M$.

- Start with best assignment through Hungarian algorithm
- Start methodically “tweaking” it by toggling associations in and out of assignment
- Maintain sorted list of best assignments so far
- During each iterative “sweep,” toggle associations in next best assignment
- k -best assignments found in decreasing order, one per sweep

A third data association method would be to use the **global all-neighbors (GAN) data association**, also known as the **joint probabilistic data association (JPDA)**, which uses *all* measurements within the gate of *each* predicted measurement, but weights each probabilistically in the correction step of the resulting filter. This concept is represented in the following figure as



However, to weigh each measurement probabilistically, one must know have a statistical model for the probability of missed detections, clutter, process and measurement models as for the PDAF, but also for the missed identification probabilities.

22.2 Probabilistic Data Association Filtering

In the context of target tracking, consider the following discrete-time stochastic state-space system

$$\begin{aligned}\vec{x}_k &= \mathbf{f}(\vec{x}_{k-1}, \vec{w}_{k-1}) \\ \vec{y}_k &= \mathbf{h}(\vec{x}_k, [DA]_k, \vec{v}_k)\end{aligned}\quad (22.13)$$

where $\vec{x} \in \mathbb{R}^{n_x N}$ is the stacked state vector of all known N targets with dimension n_x , $\vec{w}_k \in \mathbb{R}^{n_w N}$ is the *known* process noise, $\vec{y} \in \mathbb{R}^{n_y M}$ is the stacked measurement vector of all M measurements of dimension n_y , $[DA]_k$ is the data association event at k that specifies which measurement sub-vector originated from which target sub-vector *and* which measurements originated from the *random* clutter, and $\vec{v}_k \in \mathbb{R}^{n_v M}$ is the *known* measurement noise consisting of the error in the target-originated measurement and the clutter-originated measurements. The process and measurement noise have known PDFs and the *random* number of clutter-originated measurements has a *known* PMF. All these are assumed to be white and mutually independent. The initial state vector is assumed to have a known PDF and to be independent of the noises.

The **information set** available at time step k is $\vec{y}_{1:k}$, i.e. the cumulative set of measurements through time step k . Furthermore, an **information state** is a function of the available information set that summarizes the past of the system in a probabilistic sense, i.e. the information state can be substituted into the PDF of any present or future state-related statistic conditioned on the past data. For stochastic state-space systems, it can be shown the conditional PDF of the state, i.e. $f_{\vec{X}|\vec{Y}}(\vec{x}_k|\vec{y}_{1:k})$ is an information state if the process and measurement noise are white and mutually independent and the target detection and clutter is white, i.e. \vec{x}_k must be a partially observed Markov process. Thus, the information state $f_{\vec{X}|\vec{Y}}(\vec{x}_k|\vec{y}_{1:k})$ is sufficient to predict the PDF of every future state.

Optimal N -Target MMSE Filter

When the N is known, the optimal N -target state estimator in the presence of measurement-origin uncertainty in the sense of providing the MMSE can be obtained using the smoothing property of the expectation operation for the conditional mean of the state. This is obtained by averaging over all possible data association events, i.e.

$$\begin{aligned}\hat{\vec{x}}_k^{MMSE} &= \mathbb{E}[\vec{x}_k|\vec{y}_{1:k}] \\ &= \mathbb{E}[\mathbb{E}[\vec{x}_k|[DA]_{1:k}, \vec{y}_{1:k}]|\vec{y}_{1:k}] \\ &= \sum_{[DA]_{1:k,i} \in [\mathcal{DA}]_{1:k}} \mathbb{E}[\vec{x}_k|[DA]_{1:k,i}, \vec{y}_{1:k}] \Pr([DA]_{1:k,i}|\vec{y}_{1:k}) \\ &= \sum_{i=1}^{n_{DA,1:k}} \hat{\vec{x}}_k \Pr([DA]_{1:k,i}|\vec{y}_{1:k})\end{aligned}\quad (22.14)$$

where $[DA]_{1:k,i}$ is a single set of data association events from time step 1 to k , with a corresponding state estimate at time step k , $\hat{\vec{x}}_k$, and $[\mathcal{DA}]_{1:k}$ is the super set of $n_{DA,1:k}$ mutually exclusive and exhaustive

data association events. Implicitly, this model uses a Bayesian approach where one can compute prior and posterior probabilities.

For stochastic state-space systems, if the posterior conditional PDF is available, the MMSE estimator consists of the recursion of the posterior conditional PDF from time step $k - 1$ to k is

$$\begin{aligned} f_{\vec{X}|\vec{Y}}(\vec{x}_k|\vec{y}_{1:k}) &= \frac{1}{c} \sum_{i=1}^{n_{DA,k}} f_{Y|X,DA}(\vec{y}_k|\vec{x}_k, [DA]_{k,i}) \\ &\quad \times \int f_{\vec{X}|\vec{X}}(\vec{x}_k|\vec{x}_{k-1}) f_{\vec{X}|\vec{Y}}(\vec{x}_{k-1}|\vec{y}_{1:k-1}) d\vec{x}_{k-1} \\ &\quad \times \Pr([DA]_{k,i}) \end{aligned} \quad (22.15)$$

where c is the normalization constant and the integration is over the range of \vec{x}_k . This can be shown using Bayes' formula, the law of total probability, and the Chapman-Kolmogorov equation.

This recursion shows that the posterior conditional PDF for the optimal Bayes filter is a weighted sum of PDFs, i.e. a **PDF mixture**, conditioned on the current time data association events, $[DA]_{k,i}$, $i = 1, \dots, n_{DA,k}$, where $n_{DA,k}$ is the number of mutually exclusive and exhaustive data association events at time step k . Secondly, it shows that for the optimal Bayes filter, the exact posterior conditional PDF at $k - 1$ is available, then only the most recent data association event is needed at each time. Furthermore, it should be pointed out the the number of terms of the mixture at time k is given by the product

$$n_{DA,1:k} = \prod_{i=1}^k n_{DA,i} \quad (22.16)$$

which amounts to an exponential increase in the number of data association events in time.

Thus, for the stochastic state-space systems in the presence of measurement-origin uncertainty, one can use an optimal Bayes filter for this information state. However, some practical decisions must be made about the form of the state-space system and the clutter process in order to derive tractable equations. For example, if one assumes linear process and measurement models, Gaussian distributions for the process and measurement noise, and uniformly and independently distributed clutter, then the exact posterior conditional PDFs are Gaussian mixtures (GMs) with an exponentially growing number of terms. Thus, approximations must be made with regards to this Gaussian mixture. Two practical sub-optimal estimators based on MMSE optimality and approximate the posterior conditional PDF are the probabilistic data association filter (PDAF) and the joint probabilistic data association filter (JPDAF) which approximate the general posterior conditional PDF by a Gaussian distribution using similar linearization approximations as the EKF. One can also improve the PDAF and JPDAF by using a Gaussian mixture approximation while *limiting* the number of components in the mixture. Alternatively, one may also use particles to approximate the posterior conditional PDFs which would also exponentially grow in number. In either case, a key design parameter is how to limit this growing number of terms in the PDF mixture or the number of particles.

Probabilistic Data Association Filter

The **probabilistic data association filter (PDAF)** presented here consists of four steps: prediction, gating, data association, and correction, and has seven key assumptions. First, it assumes that only one target is

present, i.e. $N = 1$, with state $\vec{x} \in \mathbb{R}^{n_x}$, process model

$$\vec{x}_k = f(\vec{x}_{k-1}, \vec{w}_{k-1}) \quad (22.17)$$

and a target-originated measurement, $\vec{y}_k \in \mathbb{R}^{n_y}$

$$\vec{y}_k = h(\vec{x}_k, \vec{v}_k) \quad (22.18)$$

where \vec{w}_{k-1} and \vec{v}_k are zero-mean AWGN with covariances Q_{k-1} and R_k , respectively.

Second, the past information through time $k - 1$ about the target is summarized approximately by a sufficient statistic in the form of the Gaussian **a posteriori** PDF

$$f_{\vec{X}|\vec{Y}}(\vec{x}_k|\vec{y}_{1:k}) = f_N(\hat{\vec{x}}_{k-1|k-1}, P_{k-1|k-1}) \quad (22.19)$$

Third, it assumes that the track has been initialized, typically through some detection algorithm in the signal processing. Fourth, the target detections occur independently over time with known probability P_D . Fourth, an ellipsoidal gate is set up at each time step centered on the predicted measurement to select the candidate measurement set for data association to the target. Fifth, if the target was detected and the corresponding measurement fell in the gate, then, at most one of the gated measurements can be target-originated. Sixth, the remaining measurements are assumed to be due to clutter are independent and identically distributed with uniform spatial distribution with the number of clutter points distributed as a **spatial Poisson process** with **clutter spatial density**, $\lambda_C = \frac{\Lambda_C}{V_\phi}$, where Λ_C is the **Poisson clutter rate** and V_ϕ is the surveillance volume, or a **diffuse prior**, i.e. all observed measurements equally likely.

The **PDAF prediction step** at time step k computes the prior information state mean as

$$\hat{\vec{x}}_{k|k-1} = f(\hat{\vec{x}}_{k-1|k-1}, 0) \quad (22.20)$$

with predicted measurement

$$\hat{\vec{y}}_{k|k-1} = h(\hat{\vec{x}}_{k|k-1}, 0) \quad (22.21)$$

Then, one can approximate the prior information state covariance as

$$P_{k|k-1} = F_{k-1} P_{k-1|k-1} F_{k-1}^T + L_{k-1} Q_{k-1} L_{k-1}^T \quad (22.22)$$

where

$$F_{k-1} = \frac{\partial f(\vec{x}, \vec{w})}{\partial \vec{x}} \Big|_{\vec{x}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}=0} \quad (22.23)$$

and

$$L_{k-1} = \frac{\partial f(\vec{x}, \vec{w})}{\partial \vec{w}} \Big|_{\vec{x}=\hat{\vec{x}}_{k-1|k-1}, \vec{w}=0} \quad (22.24)$$

and the innovation covariance as

$$S_k = H_k P_{k|k-1} H_k^T + M_k R_k M_k^T \quad (22.25)$$

where

$$H_k = \frac{\partial h(\vec{x}, \vec{v})}{\partial \vec{x}} \Big|_{\vec{x}=\hat{\vec{x}}_{k|k-1}, \vec{v}=0} \quad (22.26)$$

and

$$M_{k-1} = \frac{\partial h(\vec{x}, \vec{v})}{\partial \vec{v}} \Big|_{\vec{x}=\hat{\vec{x}}_{k|k-1}, \vec{v}=0} \quad (22.27)$$

The **PDAF gating step** uses an ellipsoidal gate as

$$\mathcal{G}_E = \left\{ \vec{y}_k \in \mathbb{R}^{n_y} : \left(\vec{y}_k - \hat{\vec{y}}_{k|k-1} \right)^T S_k^{-1} \left(\vec{y}_k - \hat{\vec{y}}_{k|k-1} \right) \leq F_{\chi^2(n_y)}^{-1} (1 - \alpha) \right\} \quad (22.28)$$

where $F_{\chi^2(n_y)}^{-1} (1 - \alpha)$ is the **ellipsoidal gating threshold**, and $P_G = 1 - \alpha$ is the **gate probability**. This gating selects $M'_k \leq M_k$ measurements as the gated measurement subset, $\{\vec{y}_{k,i}\}_{i=1}^{M'_k}$. It can be shown that the **gate volume**, $V_{\mathcal{G},k}$, is given by

$$V_{\mathcal{G},k} = c_{n_y} \left(F_{\chi^2(n_y)}^{-1} (1 - \alpha) \right)^{n_y/2} |S_k|^{0.5} \quad (22.29)$$

and c_{n_y} is the volume of the n_y -dimensional hypersphere, e.g. $c_1 = 1$, $c_2 = \pi$, $c_3 = 4\pi/3$.

The **PDAF data association step** computes the data association event probabilities using the measurement likelihoods, the clutter spatial density, and the probabilities of detection and gating, i.e.

$$\Pr([DA]_{k,i} | \vec{y}_{1:k}) = \begin{cases} \frac{\mathcal{L}_{k,i}}{1 - P_D P_G + \sum_{j=1}^{M'_k} \mathcal{L}_{k,j}}, & i = 1, \dots, M'_k \\ \frac{1 - P_D P_G}{1 - P_D P_G + \sum_{j=1}^{M'_k} \mathcal{L}_{k,j}}, & i = 0 \end{cases} \quad (22.30)$$

where the measurement likelihoods are assumed Gaussian with some probability of detection

$$\mathcal{L}_{k,i} = C f_N(\vec{y}_{k,i}; \hat{\vec{y}}_{k|k-1}, S_k) P_D \quad (22.31)$$

and

$$C = \frac{1}{\lambda_C} \quad (22.32)$$

if one is using the spatial Poisson process clutter model with spatial density λ_C or

$$C = \frac{V_{\mathcal{G},k}}{M'_k} \quad (22.33)$$

if one is using a diffuse prior. This essentially results from the assumption that either M'_k or $M'_k - 1$ measurements are clutter-originating.

The **PDAF correction step** forms the filter gain as

$$K_k = P_{k|k-1} H_k^T S_k^{-1} \quad (22.34)$$

the combined innovation, probability-weighted average, is

$$\tilde{r}_k = \sum_{i=1}^{M'_k} \Pr([DA]_{k,i} | \vec{y}_{1:k}) \left(\vec{y}_{k,i} - \hat{\vec{y}}_{k|k-1} \right) \quad (22.35)$$

the posterior information state mean is computed as

$$\hat{\vec{x}}_{k|k} = \hat{\vec{x}}_{k|k-1} + K_k \tilde{r}_k \quad (22.36)$$

and the posterior information state covariance is computed as

$$\begin{aligned} P_{k|k} = & (1 - \Pr([DA]_{k,0} | \vec{y}_{1:k})) \left(P_{k|k-1} - K_k S_k K_k^T \right) + \Pr([DA]_{k,0} | \vec{y}_{1:k}) P_{k|k-1} \\ & + K_k \left(\sum_{i=1}^{M'_k} \Pr([DA]_{k,i} | \vec{y}_{1:k}) \left(\vec{y}_{k,i} - \hat{\vec{y}}_{k|k-1} \right) \left(\vec{y}_{k,i} - \hat{\vec{y}}_{k|k-1} \right)^T - \tilde{r}_k \tilde{r}_k^T \right) K_k^T \end{aligned} \quad (22.37)$$

where the first term is the covariance decrease due to a correct measurement update, the second term is the covariance unchanging due to no measurement being correct, and the third term is covariance increase due to the uncertainty in which of the M'_k measurements is correct.

Joint Probabilistic Data Association Filter

The **joint probabilistic data association filter (JPDAF)** extends the PDAF for $N > 1$ targets by considering the conditional probabilities of the following joint data association events

$$[DA]_k = \bigcap_{j=1}^M [DA]_{k,j t_j} \quad (22.38)$$

where $[DA]_{k,j t_j}$ is the data association event at time step k that measurement j originated from target t_j where t_j is the index of the target to which measurement j is associated in the data association event under consideration. The JPDAF has the following five key assumptions. First, N is known. Second, the measurements from a target can fall in the gate of another target over several time steps. Third, the past information is summarized by approximate conditional means and covariances for each target which allows the data association probabilities to be computed only for the latest set of measurements. Fourth, the states are assumed to be Gaussian distributed. Fifth, each target has a process and measurement model which do not have to be identical.

Just as for the PDAF, the **JPDAF data association step**, has two possible clutter models. The first is the Poisson PMF with clutter rate Λ_C for the number of clutter, ϕ , i.e.

$$p_\phi(\phi) = e^{-\Lambda_C} \frac{(\Lambda_C)^\phi}{\phi!} = e^{-\lambda_C V_\phi} \frac{(\lambda_c V_\phi)^\phi}{\phi!} \quad (22.39)$$

Then, the joint data association probabilities can be modeled as

$$\Pr([DA]_k | \vec{y}_{1:k}) = \frac{1}{c} \prod_j \left(\lambda_C^{-1} f_{t_j}(\vec{y}_{k,j}) \right) \prod_t (P_{D,t})^{\delta_t} (1 - P_D)^{1 - \delta_t} \quad (22.40)$$

where

$$f_{t_j}(\vec{y}_{k,j}) = f_N \left(\vec{y}_{k,j}; \hat{\vec{y}}_{k|k-1, t_j}, S_{k, t_j} \right) \quad (22.41)$$

$\hat{y}_{k|k-1}^{t_j}$ is the predicted measurement for target t_j with associated innovation covariance $S_k^{t_j}$, $P_{D,t}$ is the detection probability of target j , c is the normalization constant, τ_j is the target detection indicator function, i.e.

$$\tau_j = \begin{cases} 1, & \text{if } t_j > 0 \\ 0, & \text{if } t_j = 0 \end{cases} \quad (22.42)$$

which indicates if measurement j is associated with *any* target, and δ_t is the measurement data association indicator function, i.e.

$$\delta_t = \begin{cases} 1, & \text{if } t_j = t \text{ for some } j \\ 0, & \text{if } t_j \neq t \text{ for all } j \end{cases} \quad (22.43)$$

which indicates if *any* measurement is associated with target t .

The second is the diffuse prior i.e.

$$p_\phi(\phi) = \epsilon \text{ for all } \phi \quad (22.44)$$

Then, the joint data association probabilities can be modeled as

$$\Pr([DA]_k | \vec{y}_{1:k}) = \frac{1}{c} \phi! \prod_j \left(V_\phi f_{t_j}(\vec{y}_{k,j}) \right) \prod_t (P_{D,t})^{\delta_t} (1 - P_{D,t})^{1-\delta_t} \quad (22.45)$$

where V_ϕ is the volume of the region in which measurements not associated with a target are uniformly distributed, ϕ is the number of clutter in event $[DA]$, and c is the normalization constant. Here the term $\frac{\phi!}{V_\phi^\phi}$ substitutes for λ_C^ϕ .

If one can assume the target states conditioned on the past measurements are mutually independent, then one requires the marginal data association probabilities which are obtained from the joint probabilities by summing over all joint data association events in which the marginal data association event occurs, i.e.

$$\Pr([DA]_{k,jt} | \vec{y}_{1:k}) = \sum_{[DA]: DA_{jt} \in [DA]} \Pr([DA]_k | \vec{y}_{1:k}) \quad (22.46)$$

which allows one to decouple the JPDAF correction step for each target, $t = 1, \dots, N$, and are exactly the same as the PDAF.

However, as multiple targets may share measurements in their gates across several time steps, the estimation errors become statistically dependent. This can be taken into account by calculating state cross-covariances by using the N targets as a single stacked state vector in the correction step of the JPDAF. This additional consideration in the JPDAF results in the **joint probabilistic data association coupled filter (JPDAFC)**.

For the **JPDAFC data association step**, the conditional probability for a joint data association event is modeled as

$$\Pr([DA]_k | \vec{y}_{1:k}) = \frac{1}{c} \frac{\phi!}{M'_k!} p_C(\phi) V_\phi^{-\phi} f_{t_{j1}, t_{j2}, \dots}(\vec{y}_{k,j}, j : \tau_j = 1) \prod_t (P_{D,t})^{\delta_t} (1 - P_{D,t})^{1-\delta_t} \quad (22.47)$$

where $f_{t_{j1}, t_{j2}, \dots}$ is the joint PDF of the measurements of the targets under consideration, t_{j1} is the target to which $\vec{y}_{k,j1}$ is associated in data association event, $[DA]_k$, and c is the normalization constant. These joint probabilities are used directly in the coupled filter.

Then, for the **JPDACF correction step**, the stacked prior state vector is

$$\hat{\mathbf{x}}_{k|k-1} = \begin{bmatrix} \hat{x}_{k|k-1,1} \\ \vdots \\ \hat{x}_{k|k-1,N} \end{bmatrix} \quad (22.48)$$

The stacked predicted measurement is

$$\hat{\mathbf{y}}_{k|k-1} = \begin{bmatrix} h_1(\hat{x}_{k|k-1,1}, 0) \\ \vdots \\ h_N(\hat{x}_{k|k-1,N}, 0) \end{bmatrix} \quad (22.49)$$

The stacked measurement vector is

$$\vec{\mathbf{y}}_{k,[DA]} = \begin{bmatrix} \vec{y}_{k,j1([DA])} \\ \vdots \\ \vec{y}_{k,jN([DA])} \end{bmatrix} \quad (22.50)$$

where $j_t([DA])$ is the index of the measurement associated with target t in data association event $[DA]_k$. The stacked combined innovation is

$$\tilde{\mathbf{r}}_k = \sum_{[DA]_k} \Pr([DA]_k | \vec{\mathbf{y}}_{1:k}) (\vec{\mathbf{y}}_{k,[DA]} - \hat{\mathbf{y}}_{k|k-1}) \quad (22.51)$$

The stacked measurement matrix is

$$\mathbf{H}_k = \begin{bmatrix} H_{k,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & H_{k,N} \end{bmatrix} \quad (22.52)$$

The stacked measurement noise covariance is

$$\mathbf{R}_k = \begin{bmatrix} R_{k,1} & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & R_{k,N} \end{bmatrix} \quad (22.53)$$

The stacked prior covariance matrix is

$$\mathbf{P}_{k|k-1} = \begin{bmatrix} P_{k|k-1,11} & \cdots & P_{k|k-1,1N} \\ \vdots & \ddots & \vdots \\ P_{k|k-1,1N} & \cdots & P_{k|k-1,NN} \end{bmatrix} \quad (22.54)$$

where the cross-covariance matrices are initially set to zero. The stacked innovation covariance matrix is

$$\mathbf{S}_k = \mathbf{H}_k \mathbf{P}_{k|k-1} \mathbf{H}_k^T + \mathbf{R}_k \quad (22.55)$$

The stacked filter gain is

$$\mathbf{K}_k = \mathbf{P}_{k|k-1} \mathbf{H}_k^T \mathbf{S}_k^{-1} \quad (22.56)$$

The stacked posterior information state mean is

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + \mathbf{K}_k \tilde{\mathbf{r}}_k \quad (22.57)$$

The stacked posterior information state covariance is

$$\begin{aligned} P_{k|k} = & (1 - \Pr([DA]_{k,0} | \vec{y}_{1:k})) (\mathbf{P}_{k|k-1} - \mathbf{K}_k \mathbf{S}_k \mathbf{K}_k^T) + \Pr([DA]_{k,0} | \vec{y}_{1:k}) \mathbf{P}_{k|k-1} \\ & + \mathbf{K}_k \left(\sum_{[DA]_k} \Pr([DA]_k | \vec{y}_{1:k}) (\vec{y}_{k,[DA]} - \hat{\mathbf{x}}_{k|k-1}) (\vec{y}_{k,[DA]} - \hat{\mathbf{x}}_{k|k-1})^T - \tilde{\mathbf{r}}_k \tilde{\mathbf{r}}_k^T \right) \mathbf{K}_k^T \end{aligned} \quad (22.58)$$

References

For more information, please refer to the following

- Bar-Shalom, Y., Daum, F., and Huang, J., “The Probabilistic Data Association Filter,” *IEEE Control Systems Magazine*, Vol. 29, Issue 6, 2009, pp. 82-100
- Musicki, D., and Evans, R. J., “Joint integrated probabilistic data association: JIPDA,” in *IEEE Transactions on Aerospace and Electronic Systems*, Volume 40, Issue 3, 2004, pp. 1093-1099

22.3 Multi-Hypothesis Tracking

Multi-hypothesis tracking (MHT) is a deferred decision logic in which alternative data association hypotheses are formed whenever track-to-measurement association conflict situations occur. Then, instead of choosing the single best association hypothesis, i.e. nearest-neighbor data association, or combining all probable association hypotheses, i.e. all-neighbor data association with gating, the different association hypotheses are propagated into the future in anticipation that subsequent data will resolve the uncertainty. Thus, one seeks to find a **global hypothesis** which is a collection of compatible tracks over some number of time steps representing the estimated sequence of target states where **compatible tracks** are defined as not having any measurements in common. This also allows for the possibility of new tracks originating in surveillance volume which is not directly applicable to the JPDAF technique.

As an example, consider that two target tracks, i.e. t_1 and t_2 , with predicted states, i.e. $\hat{\mathbf{x}}_1$ and $\hat{\mathbf{x}}_2$, have been resolved before three measurements are obtained, i.e. \vec{y}_1 , \vec{y}_2 , and \vec{y}_3 , where \vec{y}_1 and \vec{y}_2 are within the gate of t_1 and \vec{y}_2 and \vec{y}_3 are with the gate of t_2 . Thus, the global hypotheses for this single time step can be written as

1. $\mathcal{H}_1: \{t_1 - \emptyset, t_2 - \emptyset, \emptyset - \vec{y}_1, \emptyset - \vec{y}_2, \emptyset - \vec{y}_3\}$
2. $\mathcal{H}_2: \{t_1 - \emptyset, t_2 - \vec{y}_2, \emptyset - \vec{y}_1, \emptyset - \vec{y}_3\}$
3. $\mathcal{H}_3: \{t_1 - \emptyset, t_2 - \vec{y}_3, \emptyset - \vec{y}_1, \emptyset - \vec{y}_2\}$
4. $\mathcal{H}_4: \{t_1 - \vec{y}_1, t_2 - \emptyset, \emptyset - \vec{y}_2, \emptyset - \vec{y}_3\}$
5. $\mathcal{H}_5: \{t_1 - \vec{y}_1, t_2 - \vec{y}_2, \emptyset - \vec{y}_3\}$

6. $\mathcal{H}_6: \{ t_1 - \vec{y}_1, \vec{t}_2 - \vec{y}_3, \emptyset - \vec{y}_2 \}$
7. $\mathcal{H}_7: \{ t_1 - \vec{y}_2, t_2 - \emptyset, \emptyset - \vec{y}_1, \emptyset - \vec{y}_3 \}$
8. $\mathcal{H}_8: \{ t_1 - \vec{y}_2, t_2 - \vec{y}_3, \emptyset - \vec{y}_1 \}$

where $t_i - \emptyset$ refers to a missed detection event and $\emptyset - \vec{y}_j$ could refer to a clutter source or a potential new target event. Thus, there are 8 feasible hypotheses, and 5 feasible tracks, i.e. 2 *a priori* tracks and 3 potential “new” tracks.

Reid's algorithm forms the basis of **hypothesis-oriented MHT (HOMHT)** which redefines the possible tracks from one time-step to the next as

$$t_i = t_i - \emptyset \quad \forall i = 1, \dots, N \quad (22.59)$$

for all $l = 1, \dots, MN$ possible data associations of N targets and M measurements

$$t_{N+l} = t_i - \vec{y}_j \quad \forall i = 1, \dots, N, j = 1, \dots, M \quad (22.60)$$

and new track hypotheses as

$$t_{N+M+N+j} = \emptyset - \vec{y}_j \quad \forall j = 1, \dots, M \quad (22.61)$$

Thus, from one time step to another, the HOMHT maintains a number K of global hypotheses across time steps and expands these multi-step global hypotheses by considering all possible track-to-measurement associations for the current time step satisfying the track compatibility requirement. However, one can see the potential combinatorial explosion in the number of global hypotheses across multiple time steps, although many of these hypotheses may be very improbable. Thus, one common method is to use Murty's algorithm to find K for all multi-step hypotheses.

However, there can be a high cost to computationally evaluating a large number of global hypotheses in HOMHT which has led to the widespread use of the alternative **track-oriented MHT (TOMHT)** which instead uses track hypothesis trees that satisfy the track compatibility requirement instead of an exhaustive list of possible global hypotheses as the methods to prune the number of track hypothesis trees are more computationally tractable. However, TOMHT does not directly provide global hypothesis probabilities as in HOMHT, but instead uses multi-dimensional assignment (MDA) using Lagrangian relaxation to implement the hypothesis output.

Track Scoring

The HOMHT yields the probability of a single cumulative joint event, i.e. a global hypothesis up to time step k , as

$$\begin{aligned} \Pr \left([DA]_{1:k}^l | \{\vec{y}_{1:k}\} \right) &= \frac{1}{c} \frac{\phi! \nu!}{M_k!} p_\phi(\phi) p_\nu(\nu) V^{-\phi-\nu} \prod_{j=1}^{M_k} [f_{t_j} (\vec{y}_{k,j})]^{\tau_j} \\ &\quad \prod_t [(P_{D_t,k})^{\delta_t} (1 - P_{D_t,k})^{1-\delta_t}] \\ &\Pr \left([DA]_{1:k-1}^s | \{\vec{y}_{1:k-1}\} \right) \end{aligned} \quad (22.62)$$

where $[DA]_{1:k}^l$ is the joint data association hypothesis l through the current time step k , $\{\vec{y}_{1:k}\}$ is the cumulative set of measurements through the current time step k , c is the normalization constant, ϕ is the

number of measurements at k deemed as false alarms in the hypothesis under consideration, ν is the number of measurements at k deemed as new targets in the hypothesis under consideration, M_k is the number of measurements at k , p_C is the PMF of the number of clutter in the current measurement set, p_ν is the PMF of the number of new targets in the current measurement set, V is the surveillance volume, f_{t_j} is the innovation PDF for the predicted measurement of track t_j to which measurement $\vec{y}_{k,j}$ is assigned in the current hypothesis, τ_j is the indicator function for the assignment of measurement $\vec{y}_{k,j}$ to a track in the hypothesis under consideration, $P_{D_{t,k}}$ is the probability of detection for target t at time step k , δ_t is the indicator function for the assignment of a measurement to track t in the hypothesis under consideration, and $[DA]_{1:k}^s$ is the parent joint data association hypothesis of $[DA]_{1:k}^l$.

Here, the whiteness of the innovation sequence of a track leads to the multiplication of the parent hypothesis probability by the PDF of the latest innovation. The factorial terms follow from a combinatorial analysis, starting with the *a priori* probabilities, that provides the correct *a posteriori* probabilities. Typically, one assumes a Poisson PMF with clutter spatial density λ_C , i.e.

$$p_\phi(\phi) = e^{-\lambda_C V} \frac{(\lambda_C V_\phi)^\phi}{\phi!} \quad (22.63)$$

and, similarly, for the number of new targets, with spatial density, λ_ν , i.e.

$$p_\nu(\nu) = e^{-\lambda_\nu V} \frac{(\lambda_\nu V_\phi)^\nu}{\nu!} \quad (22.64)$$

where V is the surveillance volume and λ_C/λ_ν are the expected number of clutter/new targets of the measurement space per time step *per unit volume*.

Using these PMFs, one has

$$\begin{aligned} \Pr([DA]_{1:k}^l | \{\vec{y}_{1:k}\}) &= \frac{1}{c} \frac{e^{-\lambda_C V} e^{-\lambda_\nu V}}{M_k!} \lambda_C^\phi \lambda_\nu^\nu \prod_{j=1}^{M_k} [f_{t_j}(\vec{y}_{k,j})]^{\tau_j} \\ &\quad \prod_t [(P_{D_{t,k}})^{\delta_t} (1 - P_{D_{t,k}})^{1-\delta_t}] \\ &\quad \Pr([DA]_{1:k-1}^s | \{\vec{y}_{1:k-1}\}) \end{aligned} \quad (22.65)$$

The marginal likelihood for track-to-measurement association which multiplies the *a priori* probability from the parent track for measurement $\vec{y}_{k,j}$ at time k having originated from track t is

$$\mathcal{L}_{k,tj}() = f_t(\vec{y}_{k,j}) P_{D_{t,k}} \quad (22.66)$$

and if it has no measurement, i.e. assigned the dummy measurement at index 0, one has

$$\mathcal{L}_{k,t0} = 1 - P_{D_{t,k}} \quad (22.67)$$

The marginal likelihood function for a clutter measurement is thus

$$\mathcal{L}_{k,\phi j} = \lambda_C \quad (22.68)$$

and the marginal likelihood function for a new target is

$$\mathcal{L}_{k,\nu j} = \lambda_\nu \quad (22.69)$$

which can both be interpreted as uniform distributions in the “average volume occupied by one such measurement.”

Thus, assuming an equal prior versus this track having no continuation, i.e. $\vec{y}_{k,j}$ is either from clutter or a new target, the probability of the event $[DA]_{k,tj}$ that measurement $\vec{y}_{k,j}$ is the continuation of track t is

$$\begin{aligned} \Pr([DA]_{k,tj} | \vec{y}_k) &= \frac{f_t(\vec{y}_{k,j}) P_{D_{t,k}}}{f_t(\vec{y}_{k,j}) P_{D_{t,k}} + (\lambda_C + \lambda_v)[1 - P_{D_{t,k}}]} \\ &= \frac{\frac{f_t(\vec{y}_{k,j}) P_{D_{t,k}}}{\lambda_C + \lambda_v}}{\frac{f_t(\vec{y}_{k,j}) P_{D_{t,k}}}{\lambda_C + \lambda_v} + 1 - P_{D_{t,k}}} \\ &= \frac{\Lambda_{k,tj}}{\Lambda_{k,tj} + 1 - P_{D_{t,k}}} \end{aligned} \quad (22.70)$$

where $\Lambda_{k,tj}$ is the likelihood ratio (LR) of track t continuing with measurement $\vec{y}_{k,j}$ versus being “extraneous,” i.e.

$$\Lambda_{k,tj} = \lambda_{ex}^{-1} f_t(\vec{y}_{k,j}) P_{D_{t,k}} \quad (22.71)$$

where

$$\lambda_{ex} = \lambda_C + \lambda_v \quad (22.72)$$

Regardless of the choice of MHT, the hypothesis evaluation process relies on the ability to compute the **log-likelihood ratio (LLR)**, also known as the **track score**, for each individual track t -to-measurement j association (T2MA) hypothesis, $\ell_{k,tj}$, at each time step k , relative to the probability of being extraneous, i.e. either clutter or a new target, as

$$\ell_{k,tj} = \ln \Lambda_{k,tj} = \ln \frac{P_{t,j,k}}{P_{ex}} \quad (22.73)$$

Furthermore, if the innovations in a track are white, then the joint likelihood is the product of the marginal likelihoods, and the track score can be accumulated from $l = 1, \dots, k$ as

$$\ell_{1:k,tj}() = \sum_{l=1}^k \ell_{l,tj(t,l)} \quad (22.74)$$

where $\ell_{l,tj(t,l)}$ is the track score of associating to target t at time l measurement $j(t,l)$. Thus, for updating the information of the different data association hypotheses, one need only update the track score for each track.

To initialize the track scores for a new track, one typically sets the mean as the pseudomeasurement mapped to the target state domain with covariance corresponding to the mapped measurement noise covariance. However, if the state is not directly observable, then one typically initializes the state to the average and the standard deviation to half the maximum expected velocity. Lastly, one can note that the probability of the T2MA can be written as

$$P_{t,j,1:k} = \frac{\exp(\ell_{tj}(1:k))}{1 + \exp(\ell_{tj}(1:k))} \quad (22.75)$$

which can be used to determine thresholds to confirm/prune hypotheses with very high/low track scores.

Multi-Dimensional Assignment

With this track score in mind, TOMHT only hypothesizes that measurements come from old tracks or are clutter, i.e. $\lambda_v = 0$. Then, any hypothesized clutter measurements are initiated as new tracks. This “two-way” measurement origin approach, as opposed allows TOMHT to use multi-dimensional assignment (MDA) which seeks to find the most likely set of tracks by maximizing the sum of all track scores while also maintaining the constraint that no tracks in the hypothesis share measurements, i.e. MDA seeks the best global hypothesis. To describe this assignment problem at time step k , consider one has N lists of hypotheses from the previous N time steps, where time step $k - N$ has confirmed tracks to that point as the root nodes of each T2MA tree, followed by the most recent $N - 1$ lists. Each item from each list should only be assigned once into a complete N -tuple consisting of one item from each of the N lists. However, to have a complete N -tuple, each list also must have a “dummy” element which stands for “clutter” or “missed detection” and has no constraints on how many times it can be assigned. For $N > 2$, MDA is an NP-hard problem and Lagrangian relaxation with approximate linear programming provides a near-optimal solution. Then, after finding the best global hypothesis, one typically implements N -scan pruning which eliminates any branches of tracks not included in the solution starting from $k - N$, thus, confirming the tracks from step $k - N$ to $k + 1 - N$ for the next window. One typically also prunes any tracks with too low of track scores.

The basic principle of the Lagrangian relaxation approach is to replace constraints, e.g. a measurement can be used by at most a single track, by Lagrange multipliers in the objective function, e.g. the sum of track scores, used in the maximization. This method involves the proper choice of Lagrange multipliers so that the solution formed from maximizing the objective function approaches the best feasible solution, in which each measurement is used by at most a single track. This optimization is complex and requires sophisticated mathematics, one can summarize the basic principles as follows. Two solutions to the hypothesis formation problem are obtained with cost defined to be the negative of score. The first solution, defined to be the relaxed or dual solution, may not satisfy the constraints, i.e. a measurement should be used once and only once. However, Lagrange multipliers are introduced into this solution and are chosen so that constraint violations are, effectively, given high costs. Thus, the number of constraint violations should be reduced over time with successive iterations of the method. A second solution, denoted the recovered or primal solution, is obtained from the dual solution by enforcing the constraints.

For example, one method for obtaining this solution starts with the assignment of the first two scans of data that was obtained by the dual solution. Then, it adds measurements from the later scans by solving an assignment matrix, that enforces the constraints, on each later scan. Thus, a feasible, but likely sub-optimal, solution is obtained. The costs of the dual solution, $q(\underline{u})$, where \underline{u} represents the Lagrangian multipliers, and the primal solution, $v(\bar{z})$, represent bounds on the cost, $v(z)$, of the true, but unknown, solution, i.e.

$$q(\underline{u}) \leq v(z) \leq v(\bar{z}) \quad (22.76)$$

where z and \bar{z} are the set of binary variables that define which tracks are included in the true and the primal solutions, respectively. Successive iterations are performed by using updated Lagrange multipliers in an attempt to increase $q(\underline{u})$ and decrease $v(\bar{z})$ and a stopping rule is defined so that the feasible primary solution is accepted when $q(\underline{u})$ and $v(\bar{z})$ are “close enough,” or the max iterations are reached.

22.4 Uncoupled-Coordinate Motion Models

This section will consider uncoupled-coordinate motion models where the *random* control input is typically taken to be the target acceleration which is the most natural choice from Newtonian dynamics as acceleration is directly related to the net force acting on the target. However, for some targets, particularly agile targets, it is more convenient to use the target jerk as the control input. Although an acceleration model can always be obtained from a jerk model by integration, one may choose a jerk model if the target motion model would be simpler. Regardless of this choice, one typically models the *unknown* control input as one of three random process models: a white noise process model, a Markov process model, and a semi-Markov Jump process model. In this section, one each dynamics model is written for a single coordinate state vector and noise vector which would be stacked for each uncoupled-coordinate in application, e.g. full two-dimensional or three-dimensional vehicle motion. These models apply to any particular vehicle tracking application.

Each model will be represented for both a continuous-time LTI state-space model, i.e.

$$\dot{\vec{x}}(t) = A \vec{x}(t) + B \vec{w}(t) \quad (22.77)$$

as well as a discrete-time LTI state-space model, i.e.

$$\vec{x}_{k+1} = F \vec{x}_k + G \vec{w}_k \quad (22.78)$$

obtained using discretization techniques with a time interval, $\Delta t = t(k+1) - t(k)$. It should be noted that the continuous-time model is more accurate than its discrete-time counterpart for most practical situations since a target moves continuously over time, but the discrete-time model is simpler to implement, though higher fidelity numerical integration can be done.

As an aside, as any continuous target trajectory can be approximated by an n^{th} polynomial to any arbitrary accuracy, the highest fidelity uncoupled-coordinate model would be a continuous-time n^{th} -order **general polynomial motion model** for the target trajectory, i.e.

$$\vec{x}(t) = \begin{bmatrix} x(t) \\ y(t) \\ z(t) \end{bmatrix} = \begin{bmatrix} a_0 & a_1 & \cdots & a_n \\ b_0 & b_1 & \cdots & b_n \\ c_0 & c_1 & \cdots & c_n \end{bmatrix} \begin{bmatrix} 1 \\ t \\ \vdots \\ t^n \end{bmatrix} + \begin{bmatrix} w_x(t) \\ w_y(t) \\ w_z(t) \end{bmatrix} \quad (22.79)$$

However, such models are usually good for data-fitting/smoothing, not the prediction/correction of target tracking. However, one can view most of the models here as special cases of this general polynomial model with *different* noise models for $w(t)$.

White Noise Motion Models

The simplest model is the **white noise acceleration model** which uses a state vector as

$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \end{bmatrix} \quad (22.80)$$

and can be written in continuous-time as

$$A = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix} \quad (22.81)$$

and

$$B = \begin{bmatrix} 0 \\ 1 \end{bmatrix} \quad (22.82)$$

and the zero-mean white process noise, $\vec{w}(t)$, models *random* acceleration with variance σ_w^2 .

The white noise acceleration model is typically written in discrete-time with

$$F = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \quad (22.83)$$

and

$$G = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \end{bmatrix} \quad (22.84)$$

where the zero-mean white noise sequence \vec{w}_k multiplied by the noise gain has a covariance

$$\text{Cov}(G\vec{w}_k) = Q = \sigma_w^2 \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 \\ \frac{1}{2}\Delta t^3 & \Delta t^2 \end{bmatrix} \quad (22.85)$$

It should be noted that if σ_w^2 is relatively small, this is also known as the “nearly” **constant-velocity (CV) model**. The effectiveness of this model depends greatly on the determination of a suitable σ_w^2 for the target which is often unknown. Thus, this parameter is often estimated using adaptive Kalman filter (AKF) techniques for the process noise covariance, e.g. based on the measurement residual statistics.

The second simplest model is the **white noise jerk model**, also known as the **Wiener process acceleration model**, which uses a state vector as

$$\vec{x}(t) = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad (22.86)$$

and can be written in continuous-time as

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \quad (22.87)$$

and

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (22.88)$$

and the zero-mean white process noise, $w(t)$, models *random* jerk with auto-correlation $\sigma_w^2\delta(h)$.

The white noise jerk model is typically written in discrete-time with

$$F = \begin{bmatrix} 1 & \Delta t & \frac{1}{2}\Delta t^2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 0 \end{bmatrix} \quad (22.89)$$

and

$$G = I_{3 \times 3} \quad (22.90)$$

where the zero-mean white process noise \vec{w}_k has covariance

$$\text{Cov}(G\vec{w}_k) = Q = \sigma_w^2 \begin{bmatrix} \frac{1}{20}\Delta t^5 & \frac{1}{8}\Delta t^4 & \frac{1}{6}\Delta t^3 \\ \frac{1}{8}\Delta t^4 & \frac{1}{3}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{6}\Delta t^3 & \frac{1}{2}\Delta t^2 & \Delta t \end{bmatrix} \quad (22.91)$$

It should be noted that if σ_w^2 is relatively small, this is also known as the “nearly” **constant-acceleration (CA) model**.

An alternative to this representation is the **Wiener sequence acceleration model** which changes the discrete-time noise gain matrix to

$$G = \begin{bmatrix} \frac{1}{2}\Delta t^2 \\ \Delta t \\ 1 \end{bmatrix} \quad (22.92)$$

and the noise covariance to

$$\text{Cov}(G\vec{w}_k) = Q = \sigma_w^2 \begin{bmatrix} \frac{1}{4}\Delta t^4 & \frac{1}{2}\Delta t^3 & \frac{1}{2}\Delta t^2 \\ \frac{1}{2}\Delta t^3 & \frac{1}{2}\Delta t^2 & \Delta t \\ \frac{1}{2}\Delta t^2 & \Delta t & 1 \end{bmatrix} \quad (22.93)$$

Markov Process Motion Models

The white noise models presented previously are simple in form, but crude approximations as actual maneuvering targets seldom have “nearly” constant velocities or accelerations that are uncoupled across coordinate directions. For example, the Wiener process model assumes the acceleration increment is independent across different time intervals which is hardly justifiable, except for its mathematical simplicity. Thus, a more realistic acceleration/jerk model would depend on *at least* its previous value. Thus, one typically considers the use of Markov process models whenever white noise models are not good enough. The **Singer acceleration model** uses a zero-mean, first-order Markov process acceleration model for the acceleration with auto-correlation

$$R_a(\Delta t) = \sigma_w^2 \exp\left(\frac{|\Delta t|}{\tau_m}\right) \quad (22.94)$$

where τ_m is **maneuver time constant** and σ^2 is the instantaneous variance of the acceleration as a random variable.

Thus, a full model defines the state vector as

$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad (22.95)$$

and can be written in continuous-time as

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & -\frac{1}{\tau_m} \end{bmatrix} \quad (22.96)$$

and

$$B = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (22.97)$$

and the zero-mean white noise driving function, $\vec{w}(t)$, models *random* acceleration with variance $2\frac{\sigma_w^2}{\tau}$.

The Singer acceleration model is typically written in discrete-time with

$$F = \begin{bmatrix} 1 & \Delta t & \tau_m^2 \left(-1 + \frac{\Delta t}{\tau_m} + \exp\left(-\frac{\Delta t}{\tau_m}\right) \right) \\ 0 & 1 & \tau_m \left[1 - \exp\left(-\frac{\Delta t}{\tau_m}\right) \right] \\ 0 & 0 & \exp\left(-\frac{\Delta t}{\tau_m}\right) \end{bmatrix} \quad (22.98)$$

and

$$G = I_{3 \times 3} \quad (22.99)$$

where the zero-mean white process noise \vec{w}_k has covariance

$$\text{Cov}(Gw_k) = Q = 2\frac{\sigma_w^2}{\tau_m} \begin{bmatrix} Q_{11} & Q_{12} & Q_{13} \\ Q_{12} & Q_{22} & Q_{23} \\ Q_{13} & Q_{23} & Q_{33} \end{bmatrix} \quad (22.100)$$

where

$$Q_{11} = \frac{1}{2}\tau_m^5 \left(1 - \exp\left(-2\frac{\Delta t}{\tau_m}\right) + 2\frac{\Delta t}{\tau_m} - 2\frac{\Delta t^2}{\tau_m^2} + \frac{2}{3}\frac{\Delta t^3}{\tau_m^3} - 4\frac{\Delta t}{\tau_m} \exp\left(-\frac{\Delta t}{\tau_m}\right) \right) \quad (22.101)$$

$$Q_{12} = \frac{1}{2}\tau_m^4 \left(\exp\left(-2\frac{\Delta t}{\tau_m}\right) + 1 - 2\exp\left(-\frac{\Delta t}{\tau_m}\right) + 2\frac{\Delta t}{\tau_m} \exp\left(-\frac{\Delta t}{\tau_m}\right) - 2\frac{\Delta t}{\tau_m} + \frac{\Delta t^2}{\tau_m^2} \right) \quad (22.102)$$

$$Q_{13} = \frac{1}{2}\tau_m^3 \left(1 - \exp\left(-2\frac{\Delta t}{\tau_m}\right) - 2\frac{\Delta t}{\tau_m} \exp\left(-\frac{\Delta t}{\tau_m}\right) \right) \quad (22.103)$$

$$Q_{22} = \frac{1}{2}\tau_m^3 \left(4\exp\left(-\frac{\Delta t}{\tau_m}\right) - 3 - \exp\left(-2\frac{\Delta t}{\tau_m}\right) + 2\frac{\Delta t}{\tau_m} \right) \quad (22.104)$$

$$Q_{23} = \frac{1}{2}\tau_m^2 \left(\exp\left(-2\frac{\Delta t}{\tau_m}\right) + 1 - 2\exp\left(-\frac{\Delta t}{\tau_m}\right) \right) \quad (22.105)$$

$$Q_{33} = \frac{1}{2}\tau_m \left(1 - \exp\left(-2\frac{\Delta t}{\tau_m}\right) \right) \quad (22.106)$$

The effectiveness of the Singer acceleration model relies on an accurate determination of the parameters: τ_m and σ^2 . For aircraft, $\tau \approx 60$ for a lazy turn and $\tau \approx 10$ to 20 for an evasive maneuver. The parameter σ^2 often is designed with a **ternary-uniform mixture distribution** defined as: the target may move without acceleration with probability P_0 , the target may accelerate or decelerate at a maximum/minimum rates $\pm a_{max}$ with equal probability P_{max} , and the target may accelerate or decelerate at a rate uniformly distributed over $(-a_{max}, a_{max})$. Then, one can show

$$\sigma^2 = \frac{a_{max}^2}{3} (1 + 4P_{max} - P_0) \quad (22.107)$$

It should also be noted that as τ_m increases, relative to Δt , the Singer acceleration model reduces to the white noise jerk model and as the τ_m decreases, relative to Δt , the Singer model acceleration model reduces to the white noise acceleration model. The Singer acceleration model is a standard model for target maneuvers as it was the first model that characterized the unknown target acceleration as a time-correlated stochastic process and served as the basis for the further development of target motion models.

In essence, the Singer acceleration model is an *a priori* model as it does not use online information about the target maneuver. As a consequence, the Singer model is symmetric as the ternary-uniform mixture distribution is symmetric which is the main shortcoming of the Singer model, i.e. the target acceleration has *zero-mean* at any moment. Notably, this is a very reasonable *a priori* assumption, but other methods may use on-line information, e.g. the mean-adaptive acceleration model, also known as the “current” model. If the target acceleration is oscillatory, a zero-mean, second-order Markov process acceleration model will serve much better than the Singer acceleration model. Furthermore, it should be noted that a Singer jerk model and non-zero-mean Singer jerk model can also be derived in a similar fashion to the Singer acceleration models.

Semi-Markov Jump Process Motion Models

In practice, many target maneuvers involve an acceleration with nonzero-mean, that may be reasonably assumed piecewise-constant. However, neither the time intervals over which the acceleration mean is piecewise-constant nor the corresponding constant levels of the nonzero mean are known to a tracker. One of the simplest models for such a piecewise-constant random process is a **semi-Markov jump process**. It differs from a Markov jump process only in that the time it stays in a mode, i.e. its **sojourn-time**, is random, i.e. it jumps at random times, while a Markov jump process has jumps at deterministic times. For example, consider the unknown input $u(t)$ to be the non-zero mean of the acceleration with *possible* mean quantized into n known levels, $\bar{a}_1, \dots, \bar{a}_n$. Then, the sequence $u(t_0), \dots, u(t_k), \dots$ among these level is a semi-Markov process with known transition probabilities

$$\Pr\{u(t_k) = \bar{a}_j | u(t_{k-1}) = \bar{a}_i\} \quad \text{for } i, j = 1, \dots, n \quad (22.108)$$

and the sojourn-time PMFs are

$$P_{ij}(t) = \Pr(\tau_{ij} \leq t) \quad (22.109)$$

where $\tau_{ij} = t_k - t_{k-1}$ is the sojourn-time in mode \bar{a}_i before it jumps to mode \bar{a}_j .

Then, one such model is the **Markov jump-mean acceleration model** which models the target acceleration as the Singer acceleration $(\tilde{a})(t)$, with *non-zero mean* that is a semi-Markov process, i.e.

$$\ddot{x}(t) = -\beta \dot{x}(t) + u(t) + (\tilde{a})(t) \quad (22.110)$$

where β is a drag coefficient. Thus, this model defines the state vector as

$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad (22.111)$$

and can be written as a continuous-time state-space model as

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & -\beta & 1 \\ 0 & 0 & -\frac{1}{\tau_m} \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \quad (22.112)$$

where the zero-mean white noise driving function, $\vec{w}(t)$, models *random* acceleration with variance $2\frac{\sigma_w^2}{\tau}$. The unknown acceleration mean is typically estimated by a weighted sum of the quantization levels, i.e.

$$\hat{u}(t) = \sum_{i=1}^n \bar{a}_i \Pr\{u(t) = \bar{a}_i | y(s), s \leq t\} \quad (22.113)$$

where $\vec{y}(s)$, $s \leq t$ denotes all measurements up to time s . Three important issues associated with this model are the design of the input modes, the transition probabilities, and the sojourn-time.

22.5 Turning Motion Models

Coupled-coordinate models are based on the target kinematics, in contrast to those of uncoupled-coordinate motion based on random processes due to the fact random processes are more natural for modeling time correlation than kinematics for modeling spatial correlation. Coordinate-coupled target models are highly dependent on the choice of the state components, e.g. to describe two-dimensional motion, the state vector is at least four-dimensional. The choice of the state components and kinematic model is a non-trivial with consideration given to the expected target trajectory with random perturbation, the accuracy of the unavoidable approximations in the model, and the sensor coordinate system. This section will focus on kinematic-based target models for maneuvering flight vehicles, in particular, **turning motion models** which can be applied to both ground, surface, and air vehicles.

As an aside, in some applications, the decision for using a non-maneuvering or maneuvering motion model can be made *a priori* or the problem can be posed to the target tracking system as it operates in real-time. This decision may be performed by an **adaptive tracker** which estimates the unknown model parameters, often noise, alongside the state. Secondly, the system may use a **maneuver detector** which attempts to determine the **onset time** of any maneuvers, e.g. using χ -squared tests or GLRTs. Alternatively, the system may also use set-based methods known as **multi-model target tracking (MM-TT)** where one uses the full model set of all considered maneuvers for the vehicle(s) to be tracked in the target tracking algorithm.

Two-Dimensional Coordinated-Turn Models

For two-dimensional motion, the **standard curvilinear motion model** can be written from kinematics as

$$\begin{aligned} \dot{x}(t) &= v(t) \cos \psi(t) \\ \dot{y}(t) &= v(t) \sin \psi(t) \\ \dot{v}(t) &= a_t(t) \\ \dot{\psi}(t) &= \frac{a_n(t)}{v(t)} \end{aligned} \quad (22.114)$$

where (x, y) are the target position in Cartesian coordinates, v is the speed, ψ is the heading angle, a_t is the target tangential acceleration, also known as the “along-track” acceleration, and a_n is the target normal acceleration, also known as the “cross-track” acceleration. These two accelerations are then treated as the unknown inputs to the target motion model.

Three special cases exist for this general curvilinear motion. If $a_t = a_n = 0$, then one has rectilinear, constant velocity (CV) motion. If $a_t \neq 0$, $a_n = 0$, then one has rectilinear, acceleration motion, e.g. if $a_t = \text{constant}$ then constant acceleration (CA) motion. For both cases, previous uncoupled-coordinate motion models apply. However, if $a_t = 0$ and $a_n \neq 0$, then one has a **coordinated-turn (CT) motion models**, i.e. constant-speed, constant-turn-rate motion and can be simplified to two simpler models specified in terms of the **turn-rate**, $\omega = \dot{\phi}$, which may be known or unknown.

The **coordinated-turn with known turn-rate (CT-KTR) model** assumes ω is known, uses a four-dimensional state vector as

$$\vec{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix} \quad (22.115)$$

and can be written in as a continuous-time LTI state-space model, i.e.

$$\dot{\vec{x}} = A_{ct}(\omega) \vec{x} + \vec{w}(t) \quad (22.116)$$

where

$$A_{ct}(\omega) = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \omega & 0 \\ 0 & 0 & 0 & -\omega \end{bmatrix} \quad (22.117)$$

The CT-KTR model can be discretized as

$$\vec{x}_{k+1} = F_{ct}(\omega) \vec{x}_k + \vec{w}_k \quad (22.118)$$

$$F_{ct} = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega\Delta t)}{\omega} & -\frac{1-\cos(\omega\Delta t)}{\omega} \\ 0 & 1 & \frac{1-\cos(\omega\Delta t)}{\omega} & \frac{\sin(\omega\Delta t)}{\omega} \\ 0 & 0 & \cos(\omega\Delta t) & -\sin(\omega\Delta t) \\ 0 & 0 & \sin(\omega\Delta t) & \cos(\omega\Delta t) \end{bmatrix} \quad (22.119)$$

or approximated for $\omega\Delta t \approx 0$ by

$$F_{ct} \approx \begin{bmatrix} 1 & 0 & \Delta t & -\frac{1}{2}\omega^2\Delta t^2 \\ 0 & 1 & \frac{1}{2}\omega^2\Delta t^2 & \Delta t \\ 0 & 0 & 1 - \frac{1}{2}\omega^2\Delta t^2 & -\omega\Delta t \\ 0 & 0 & \omega\Delta t & 1 - \frac{1}{2}\omega^2\Delta t^2 \end{bmatrix} \quad (22.120)$$

and $\vec{w}(t)$ is zero-mean AWGN used to perturb the model from the ideal CT motion by some amount. This is only a decent model if the constant turn-rate is known approximately *a priori*. However, this model is often utilized with adaptive approaches to learn ω and with multiple-model methods using Markov or semi-Markov sequences for different discrete values of ω .

The **coordinated-turn with unknown turn-rate (CT-UTR) models** assume that ω is driven by a stochastic process, typically either a Wiener process

$$\begin{aligned} \dot{\omega}(t) &= w_\omega(t) \\ \omega_{k+1} &= \omega_k + w_{\omega,k} \end{aligned} \quad (22.121)$$

or a first-order Markov process

$$\begin{aligned}\dot{\omega}(t) &= -\frac{1}{\tau_\omega} \omega(t) + w_\omega(t) \\ \omega_{k+1} &= \exp\left(-\frac{\Delta t}{\tau_\omega}\right) \omega_k + w_{\omega,k}\end{aligned}\quad (22.122)$$

where τ_ω is the correlation time constant for the turn-rate and $\vec{w}(t)$ is again some zero-mean AWGN used to perturb the model from the ideal CT motion by some amount.

Thus, the five-dimensional state vector is

$$\vec{x} = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \\ \omega \end{bmatrix} \quad (22.123)$$

and the noise vector is

$$\vec{w} = \begin{bmatrix} w_{\dot{x}} \\ w_{\dot{y}} \\ w_\omega \end{bmatrix} \quad (22.124)$$

The CT-UTR model can be written as a continuous-time LPV state-space model, i.e.

$$\dot{\vec{x}} = f(\vec{x}(t), \vec{w}(t)) = \begin{bmatrix} A_{ct}(\omega) & 0 \\ 0_{1 \times 4} & -\alpha \end{bmatrix} \vec{x} + L \vec{w}(t) \quad (22.125)$$

where $\alpha = 0$ or $\frac{1}{\tau_\omega}$, the noise gain matrix is

$$L = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (22.126)$$

which can be discretized as

$$\vec{x}_{k+1} = f(\vec{x}_k, \vec{w}_k) = \begin{bmatrix} F_{ct}(\omega) & 0 \\ 0_{1 \times 4} & \beta \end{bmatrix} \vec{x}_k + \begin{bmatrix} \frac{1}{2} \Delta t^2 & 0 & 0 \\ 0 & \frac{1}{2} \Delta t^2 & 0 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & 1 \end{bmatrix} \vec{w}_k \quad (22.127)$$

where $\beta = 1$ or $\exp\left(-\frac{\Delta t}{\tau_\omega}\right)$. As this is a nonlinear equation in ω , one must use a nonlinear filter with the CT-UTR model, e.g. an EKF. In this case, the Jacobians of the discrete-time model are notably

$$\frac{\partial f(\vec{x}, \vec{w})}{\partial \vec{x}} = \begin{bmatrix} 1 & 0 & \frac{\sin(\omega\Delta t)}{\omega} & -\frac{1-\cos(\omega\Delta t)}{\omega} & \left(\frac{\omega\Delta t \cos(\omega\Delta t)-\sin(\omega\Delta t)}{\omega^2}\right)\dot{x} - \left(\frac{\omega\Delta t \sin(\omega\Delta t)-1+\cos(\omega\Delta t)}{\omega^2}\right)\dot{y} \\ 0 & 1 & \frac{1-\cos(\omega\Delta t)}{\omega} & \frac{\sin(\omega\Delta t)}{\omega} & \left(\frac{\omega\Delta t \sin(\omega\Delta t)-1+\cos(\omega\Delta t)}{\omega^2}\right)\dot{x} + \left(\frac{\omega\Delta t \cos(\omega\Delta t)-\sin(\omega\Delta t)}{\omega^2}\right)\dot{y} \\ 0 & 0 & \cos(\omega\Delta t) & -\sin(\omega\Delta t) & -\Delta t \sin(\omega\Delta t)\dot{x} - \Delta t \cos(\omega\Delta t)\dot{y} \\ 0 & 0 & \sin(\omega\Delta t) & \cos(\omega\Delta t) & \Delta t \cos(\omega\Delta t)\dot{x} - \Delta t \sin(\omega\Delta t)\dot{y} \\ 0 & 0 & 0 & 0 & \beta \end{bmatrix} \quad (22.128)$$

and

$$\frac{\partial f(\vec{x}, \vec{w})}{\partial \vec{w}} = L_k = \begin{bmatrix} \frac{1}{2}\Delta t^2 & 0 & 0 \\ 0 & \frac{1}{2}\Delta t^2 & 0 \\ \Delta t & 0 & 0 \\ 0 & \Delta t & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (22.129)$$

Three-Dimensional Turn Models

The basic equation for three-dimensional turn motion, relates the angular velocity, $\vec{\omega}$, to the vehicle velocity and acceleration vectors as

$$\vec{\omega} = \frac{\vec{x} \times \ddot{\vec{x}}}{\vec{x}^T \vec{x}} \quad (22.130)$$

where for a three-dimensional fixed-center constant-angular-velocity (CAV) model, one has

$$\ddot{\vec{x}} = \vec{\omega} \times \vec{x} \quad (22.131)$$

Thus, defining the derivative of acceleration as

$$\ddot{\vec{x}} = -\|\vec{\omega}\|_2^2 \vec{x} = \frac{\|\vec{x} \times \ddot{\vec{x}}\|}{\vec{x}^T \vec{x}} \vec{x} \quad (22.132)$$

Thus, the three-dimensional **fixed-center constant-turn-rate (FC-CTR) model** is a second-order Markov process

$$\ddot{\vec{x}} = -\omega^2 \vec{x} + \vec{w} \quad (22.133)$$

where $\omega = \|\vec{\omega}\|_2$ and \vec{w} is white noise with variance $\sigma_\omega^2 I_{3 \times 3}$.

The FC-CTR has *for each coordinate* a state vector as

$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad (22.134)$$

and can be written as a continuous-time LTI state-space model as

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -\omega^2 & 0 \end{bmatrix} \vec{x} + \vec{w}(t) \quad (22.135)$$

which can be discretized as

$$\vec{x}_{k+1} = \begin{bmatrix} 1 & \frac{\sin(\omega\Delta t)}{\omega} & \frac{1-\cos(\omega\Delta t)}{\omega^2} \\ 0 & \cos(\omega\Delta t) & \frac{\sin(\omega\Delta t)}{\omega} \\ 0 & -\omega \sin(\omega\Delta t) & \cos(\omega\Delta t) \end{bmatrix} \vec{x}_k + \begin{bmatrix} \frac{\omega\Delta t - \sin(\omega\Delta t)}{\omega^3} \\ \frac{1-\cos(\omega\Delta t)}{\omega^2} \\ \frac{\sin(\omega\Delta t)}{\omega} \end{bmatrix} w_k \quad (22.136)$$

where $\text{Cov}(w_k) = \sigma_\omega^2 I$. Thus, the coordinates are coupled only through the common ω , which must be estimated in this form. Note that for a target with constant speed, the velocity and acceleration vectors are orthogonal, i.e. $\ddot{\vec{x}}^T \dot{\vec{x}} = 0$, then

$$\omega = \frac{\|\ddot{\vec{x}}\|_2}{\|\dot{\vec{x}}\|_2} \quad (22.137)$$

which may be imposed as a constraint on the potentially unknown $\vec{\omega}$.

Alternatively, the **constant-angular-velocity (CAV) model** uses a state vector

$$\vec{x} = \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \\ \vec{\omega} \end{bmatrix} \quad (22.138)$$

and can be written as a continuous-time nonlinear state-space model as

$$\dot{\vec{x}}(t) = \begin{bmatrix} 0 & I_{3 \times 3} & 0 \\ 0 & [\vec{\omega}]^\times & 0 \\ 0 & 0 & 0 \end{bmatrix} \vec{x}(t) + \vec{w}(t) \quad (22.139)$$

where the angular velocity can be estimated as part of the state vector. Note that if $\vec{\omega} = [0 \ 0 \ \omega]^T$, then this model reduces to the CT-KTR model.

The **nearly constant-turn-rate model (N-CTR) model** assumes the acceleration is a second-order Gauss-Markov process with state-dependent coefficients, i.e.

$$\ddot{\vec{x}} = -2\alpha \dot{\vec{x}} - (\alpha^2 + \omega^2) \vec{x} + \vec{w} \quad (22.140)$$

where $\omega = \|\vec{\omega}\|_2$ is the turn-rate,

$$\alpha = -\frac{\dot{\vec{x}}^T \ddot{\vec{x}}}{\dot{\vec{x}}^T \dot{\vec{x}}} \quad (22.141)$$

is a normalized target drag term and

$$\vec{w} = \vec{\omega}_B \times \vec{x} + \ddot{\vec{x}}_B \quad (22.142)$$

reflects the effect of the forces and moments applied to the target, which are unknown, and taken as AWGN with variance $\sigma_\omega^2 I$.

The N-CTR has *for each coordinate* a state vector as

$$\vec{x} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix} \quad (22.143)$$

and can be written as a continuous-time LTI state-space model as

$$\dot{\vec{x}} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -(\alpha^2 + \omega^2) & -2\alpha \end{bmatrix} \vec{x} + \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} w(t) \quad (22.144)$$

which can be discretized as

$$\vec{x}_{k+1} = F \vec{x}_k + G w_k \quad (22.145)$$

where

$$F = \begin{bmatrix} 1 & \frac{2\alpha\omega - \exp(-\alpha\Delta t)(2\alpha\omega\cos(\omega\Delta t) + (\alpha^2 - \omega^2)\sin(\omega\Delta t))}{\omega(\alpha^2 + \omega^2)} & \frac{\omega - \exp(-\alpha\Delta t)(\omega\cos(\omega\Delta t) + \alpha\sin(\omega\Delta t))}{\omega(\alpha^2 + \omega^2)} \\ 0 & \frac{\exp(-\alpha\Delta t)(\omega\cos(\omega\Delta t) + \alpha\sin(\omega\Delta t))}{\omega} & \frac{\exp(-\alpha\Delta t)\sin(\omega\Delta t)}{\omega} \\ 0 & -\frac{(\alpha^2 + \omega^2)\exp(-\alpha\Delta t)\sin(\omega\Delta t)}{\omega} & \frac{\exp(-\alpha\Delta t)(\omega\cos(\omega\Delta t) - \alpha\sin(\omega\Delta t))}{\omega} \end{bmatrix} \quad (22.146)$$

$$G = \begin{bmatrix} \frac{\omega(-2\alpha + (\alpha^2 + \omega^2)\Delta t) + \exp(-\alpha\Delta t)(2\alpha\omega\cos(\omega\Delta t) + (\alpha^2 - \omega^2)\sin(\omega\Delta t))}{\omega(\alpha^2 + \omega^2)^2} \\ \frac{\omega - \exp(-\alpha\Delta t)(\omega\cos(\omega\Delta t) + \alpha\sin(\omega\Delta t))}{\omega(\alpha^2 + \omega^2)} \\ \frac{\exp(-\alpha\Delta t)\sin(\omega\Delta t)}{\omega} \end{bmatrix} \quad (22.147)$$

and $\text{Cov}(w_k) = \sigma_w^2$. Thus, the coordinates are coupled only through the common ω and α .

Lastly, the **non-constant-speed coordinated-turn (NCS-CT) model** assumes the target acceleration is

$$\ddot{\vec{x}} = \vec{\omega} \times (\ddot{\vec{x}} - \vec{g}) + C_{N \leftarrow B}(\dot{\vec{x}}, \ddot{\vec{x}}) \vec{u} \quad (22.148)$$

where \vec{g} is the gravity vector and the direction cosine matrix is formed as

$$C_{N \leftarrow B}(\dot{\vec{x}}, \ddot{\vec{x}}) = \begin{bmatrix} \frac{\dot{\vec{x}}}{\|\dot{\vec{x}}\|_2} & \frac{(\ddot{\vec{x}} - \vec{g}) \times \dot{\vec{x}}}{\|(\ddot{\vec{x}} - \vec{g}) \times \dot{\vec{x}}\|_2} & \frac{\dot{\vec{x}}}{\|\dot{\vec{x}}\|_2} \times \frac{(\ddot{\vec{x}} - \vec{g}) \times \dot{\vec{x}}}{\|(\ddot{\vec{x}} - \vec{g}) \times \dot{\vec{x}}\|_2} \end{bmatrix} \quad (22.149)$$

and \vec{u} is a vector-valued zero-mean, first-order Gauss-Markov process that models random perturbations in the acceleration with respect to the body frame. The time-correlated perturbations accounts for the unmodeled motions and is modeled in the body frame to be more precise. This, however, necessitates a frame transformation which introduces a dependency of the model on velocity and acceleration estimates.

This creates a highly nonlinear model and may lead to large dynamic model errors. The continuous-time state-space form of this model is

$$\dot{\vec{x}} = \begin{bmatrix} \dot{\vec{x}} \\ \ddot{\vec{x}} \\ \frac{\dot{\vec{x}} \times \dot{\vec{x}}}{\dot{\vec{x}}^T \dot{\vec{x}}} \times (\ddot{\vec{x}} - \vec{g}) + C_{N \leftarrow B}(\dot{\vec{x}}, \ddot{\vec{x}}) \vec{u} \\ -\frac{1}{\tau_m} \vec{u} \end{bmatrix} + \begin{bmatrix} 0_{3 \times 3} \\ 0_{3 \times 3} \\ 0_{3 \times 3} \\ I_{3 \times 3} \end{bmatrix} \vec{w}_B \quad (22.150)$$

which uses a state vector

$$\vec{x} = \begin{bmatrix} \vec{x} \\ \dot{\vec{x}} \\ \ddot{\vec{x}} \\ \vec{u} \end{bmatrix} \quad (22.151)$$

22.6 Ballistic Flight Motion Models

Ballistic flight is defined as a flight path that is largely predetermined by the performance characteristics specific for type of flight vehicle which includes launch vehicles, ballistic missiles, decoys, debris, satellites, and projectiles, though some advanced ballistic missiles can undergo small maneuvers usually for re-targeting. Regardless, tracking the motion of *unknown ballistic targets (BT)* can be challenging due to the variety of uncertainties, most of which result from the target type uncertainty, the principal uncertainty in **ballistic target tracking**. The general BT flight path can be divided into three phases: boost, coast, and reentry. The **boost phase** is the endo-atmospheric, powered flight, which begins at launch and continues until thrust cutoff or burnout. The **coast phase** is the exo-atmospheric, free-motion flight which begins at thrust cutoff or burnout and continues until the atmosphere is reached again. The **reentry phase** is the endo-atmospheric, free-motion flight, which begins when the atmospheric drag becomes considerable and continues until landing or impact.

Though a general motion model based on random processes can be used for the entire flight path of a BT, those models are typically crude as opposed to taking advantage of the fact that ballistic flight can be primarily modeled by three forces: gravity, thrust, and aerodynamic, i.e. drag and possibly lift. Furthermore, not all of these primary forces significantly characterize every flight phase of BTs. The boost phase is characterized by a large thrust, which may be subject to abrupt, jump-wise changes, as well as drag and gravity. After the boost phase, drag is no longer present and thrust vanishes or drops to a very low level. Thus, the coast phase motion is characterized essentially by gravity only though small re-targeting maneuvers are possible. The reentry phase motion is characterized by a rapid drag-induced deceleration with possible lateral accelerations. In summary, the significant forces in the different phases are gravity in coast; aerodynamic and gravity in reentry; and thrust, gravity, and aerodynamic in boost. This section will discuss the relevant **ballistic flight motion models** in order from the least number of significant forces to greatest.

Lastly, this section uses \vec{a} for acceleration, \vec{v} for velocity, and \vec{p} for position where the reference frame for the coordinates are chosen as either the Earth-Centered, Inertial (ECI) or the local East-North-Up (ENU) reference frames. Thus, the kinematic portion of the state vector has the form

$$\vec{x} = \begin{bmatrix} \vec{p} \\ \vec{v} \end{bmatrix} \quad (22.152)$$

and its time derivative as

$$\dot{\vec{x}} = \begin{bmatrix} \vec{v} \\ \vec{a} \end{bmatrix} \quad (22.153)$$

and the primary consideration is how to model the acceleration \vec{a} appropriately.

Acceleration Models

The total acceleration of a BT in an inertial frame, \vec{a} , can be decomposed as

$$\vec{a} = \vec{a}_G + \vec{a}_T + \vec{a}_A = \vec{a}_G + \vec{a}_T + \vec{a}_D + \vec{a}_L \quad (22.154)$$

where \vec{a}_G is the acceleration due to gravity, \vec{a}_T is the acceleration due to thrust, \vec{a}_A is the acceleration due to aerodynamic forces, \vec{a}_D is the acceleration due to drag, and \vec{a}_L is the acceleration due to lift. In an

Earth-fixed frame, the total acceleration will also include those induced by the Earth's angular velocity, $\vec{\omega}_E$, i.e. the Coriolis and centrifugal acceleration

$$\vec{a}_C = 2[\vec{\omega}_E]_x \vec{v} + [\vec{\omega}_E]_x^2 \vec{p} \quad (22.155)$$

where \vec{v} is the target velocity and $[\bullet]_x$ is the skew-symmetric matrix. For ballistic target tracking, one typically uses a range of force models which are summarized as follows. Note that one would need to *subtract* the Earth rotation term from the following for an Earth-fixed frame.

For gravity, one may use a flat-Earth, spherical-Earth, or ellipsoidal-Earth model. A **flat-Earth gravity model** assumes a flat, non-rotating Earth and the gravity acting on the target is constant. In East-North-Up (ENU) coordinates, one has

$$\vec{a}_{G,f-E} = \begin{bmatrix} 0 \\ 0 \\ -g_0 \end{bmatrix} \quad (22.156)$$

where g_0 is the **standard gravitational acceleration**, i.e. 9.80665 m-s^{-2} .

A **spherical-Earth gravity model** assumes that Earth and the target are point masses at their centers of gravity, no other body's gravity affects either, and the target has negligible mass. Then, one can use Newton's inverse-square gravitational law as

$$\vec{a}_{G,s-E}(\vec{r}) = -\frac{-\mu_E}{\|\vec{r}\|_2^3} \vec{r} = -\frac{-\mu_E}{\|\vec{r}\|_2^2} \vec{u}_r \quad (22.157)$$

where μ_E is the Earth's gravitational constant, i.e. $398600.440 \text{ km}^3\text{-s}^{-2}$, \vec{r} is the vector between Earth's and the target's center of gravity, and \vec{u}_r is its unit vector.

An **ellipsoidal-Earth gravity model** accounts for the Earth's oblate spheroid shape by including the **second-order gravitational harmonic**, J_2 , of the Earth's gravitational field, i.e. $1.75553 \times 10^{10} \text{ km}^5\text{-s}^{-2}$ from the Joint Gravity Model (JGM-3). This can be shown to be the following difference.

$$J_2 = \frac{I_{zz,E} - I_{xx,E}}{m_E R_e^2} \quad (22.158)$$

where $I_{zz,E}$ is the polar moment of inertia of the ellipsoidal Earth, $I_{xx,E}$ is the equatorial moment of inertia of the ellipsoidal Earth, m_E is the mass of Earth, and R_e is the equatorial radius of Earth. Then, accounting for this second-order correction, one has

$$\vec{a}_{G,e-E}(\vec{r}) = -\frac{-\mu_E}{\|\vec{r}\|_2^2} \left[\vec{u}_r + \frac{3}{2} J_2 \left(\frac{R_e}{\|\vec{r}\|_2} \right)^2 \left((1 - 5(\vec{u}_r^T \vec{u}_z)^2) \vec{u}_r + 2\vec{u}_r^T \vec{u}_z \vec{u}_z \right) \right] \quad (22.159)$$

where R_e is the equatorial radius of Earth and \vec{u}_z is the unit vector along the z -axis of an Earth-centered reference frame.

For the aerodynamic forces, one typically models the acceleration due to the drag force as

$$\vec{a}_D = -\frac{1}{2m} S C_D \rho(h) \|\vec{v}_r\|_2 \vec{v}_r = -\frac{1}{2m} S C_D \rho(h) \|\vec{v}\|_2^2 \vec{u}_v \quad (22.160)$$

where m is the target mass, S is the cross-sectional surface area to the velocity, \vec{v} is the velocity relative to the atmosphere in a non-inertial frame, C_D is the drag coefficient which generally depends on S and \vec{v} ,

but is sometimes assumed constant, and $\rho(h)$ is the air density and depends on the target altitude, h , and is typically approximated analytically as

$$\rho(h) \approx \rho_0 \exp\left(-\frac{h}{h_0}\right) \quad (22.161)$$

where $\rho_0 = 1.225 \text{ kg-m}^{-3}$ and $h_0 = 10,400 \text{ m}$ are often used, though higher fidelity models may also be used. Here, \vec{u}_v is the unit vector in the direction of the velocity.

In ballistic target models, one often alternatively uses the **drag parameter** as

$$\alpha_D = \frac{SC_D}{m} \quad (22.162)$$

or the **ballistic coefficient (BC)** as

$$\beta_D = \frac{m}{SC_D} \quad (22.163)$$

Thus, the acceleration due to the drag force can be written as

$$\vec{a}_D = -\frac{1}{2}\alpha_D\rho_0 \exp\left(-\frac{h}{h_0}\right) \|\vec{v}\|_2^2 \vec{u}_v \quad (22.164)$$

For ballistic targets, lift occurs when the target velocity vector and thrust vectors are not aligned, i.e. the target is at some angle of attack, typically for the purposes of maneuvering during reentry or boosting. Furthermore, the lift during a maneuver can also be modeled as

$$\vec{a}_L = -\frac{1}{2}\rho(h)\|\vec{v}\|_2^2 \vec{a}_L \vec{u}_L \quad (22.165)$$

where modeling the two coordinates of \vec{a}_L and \vec{u}_L depend on the reference frame used for the maneuvering target. This will also add a lift-induced drag to the overall target drag. Lastly, when thrust is expected during the boost phase, one cannot observe both the drag and the thrust separately. Thus, one typically models the combined thrust and drag as the **net acceleration**, \vec{a}_{T-D} ,

$$\vec{a} = \vec{a}_G + \vec{a}_T + \vec{a}_D = \vec{a}_G + \vec{a}_{T-D} \quad (22.166)$$

This net acceleration may be taken as constant or time-varying in which case, it is modeled as an unknown stochastic process and included in the estimation procedure using basic assumptions on its form.

Coast Motion Models

In coast motion models, one typically only considers gravity as the significant force. Thus, one has for the total acceleration

$$\dot{\vec{x}} = \begin{bmatrix} \vec{v} \\ \vec{a}_G \end{bmatrix} \quad (22.167)$$

and one need only decide between the spherical-Earth or ellipsoidal-Earth models and the ECI or ENU reference frames. This provides four options for motion models. For the spherical-Earth and ECI model, one has

$$\vec{a} = \vec{a}_{G,s-E}(\vec{p}) = -\frac{\mu_E}{\|\vec{p}\|_2^3} \vec{p} \quad (22.168)$$

and the motion is governed by Kepler's equation of motion.

Though this motion is highly nonlinear, one can predict the state forward-in-time using the following algorithm. Given, $\vec{x}(t_0) = [\vec{p}_0 \ \vec{v}_0]^T$, the predicted state at time $t = t_0 + \Delta t$ is given by

$$\begin{aligned}\vec{p} &= \left(1 - \frac{1 - \lambda^2 C}{\|\vec{p}\|_0}\right) \vec{p}_0 + \left(\Delta t - \frac{\lambda^3 S}{\sqrt{\mu_E}}\right) \vec{v}_0 \\ \vec{v} &= \left(\frac{\sqrt{\mu_E} \lambda (\zeta S - 1)}{\|\vec{p}_0\|_2 \|\vec{p}\|_2}\right) \vec{p}_0 + \left(1 - \frac{\lambda^2 C}{\|\vec{p}\|_0}\right) \vec{v}_0\end{aligned}\quad (22.169)$$

where C , S , and ζ are known functions of λ , and the parameter λ is defined through $\dot{\lambda} = \sqrt{\mu_E}/\|\vec{p}\|_2$ and is typically computed using a Newton iteration scheme for the **time-of-flight (TOF) equation** which can be defined for elliptical, parabolic, and hyperbolic flight paths.

For elliptical orbits, i.e. $a = 2/\|\vec{p}_0\|_2 - \|\vec{v}_0\|_2^2/\mu_E > 0$, one can initialize the iteration on λ as

$$\begin{aligned}\alpha &= \frac{1 - a \|\vec{p}_0\|_2}{\sqrt{\mu_E}} \\ \beta &= \frac{\vec{p}_0^T \vec{v}_0}{\mu_E} \\ \gamma &= \frac{\|\vec{p}_0\|}{\sqrt{\mu_E}} \\ \lambda_0 &= \frac{a \Delta t}{\sqrt{\mu_E}}\end{aligned}\quad (22.170)$$

and iterate λ_i for $i = 1, 2, \dots$ until $|\Delta t - \tau| < \epsilon$:

$$\begin{aligned}\zeta &= a \lambda_i^2 \\ C &= \frac{1 - \cos \sqrt{\zeta}}{\zeta} \\ S &= \frac{\sqrt{\zeta} - \sin \sqrt{\zeta}}{\zeta \sqrt{\zeta}} \\ \tau &= \alpha \lambda_i^3 S + \beta \lambda_i^2 C + \gamma \lambda_i \\ \frac{d\tau}{d\lambda} &= \alpha \lambda_i^2 C + \beta \lambda_i (1 - \zeta S) + \gamma \\ \lambda_{i+1} &= \lambda_i + \left[\frac{d\tau}{d\lambda} \right]^{-1} (\Delta t - \tau)\end{aligned}\quad (22.171)$$

For the ellipsoidal-Earth and ECI model, one has

$$\vec{a} = \vec{a}_{G,e-E}(\vec{p}) \quad (22.172)$$

where

$$\vec{u}_p = \frac{\vec{p}}{\|\vec{p}\|} \quad (22.173)$$

and

$$\vec{u}_z = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (22.174)$$

Thus, one has

$$\vec{a} = -\frac{\mu_E}{\|\vec{p}\|_2^3} \begin{bmatrix} 1 + \frac{3J_2R_e^2}{2\|\vec{p}\|_2^2} \left(1 - 5 \left(\frac{z}{\|\vec{p}\|_2} \right)^2 \right) \\ 1 + \frac{3J_2R_e^2}{2\|\vec{p}\|_2^2} \left(1 - 5 \left(\frac{z}{\|\vec{p}\|_2} \right)^2 \right) \\ 1 + \frac{3J_2R_e^2}{2\|\vec{p}\|_2^2} \left(3 - 5 \left(\frac{z}{\|\vec{p}\|_2} \right)^2 \right) \end{bmatrix} \vec{p} \quad (22.175)$$

For the spherical-Earth gravity model and ENU reference frame, one has

$$\vec{a} = \vec{a}_{G,s-E}(\vec{p}) - 2\vec{\omega}_E \Phi_0 \vec{v} - \omega_E^2 \Phi_0^2 \vec{x} \quad (22.176)$$

where

$$\Phi_0 = \begin{bmatrix} 0 & -\sin \phi_0 & \cos \phi_0 \\ \sin \phi_0 & 0 & 0 \\ -\cos \phi_0 & 0 & 0 \end{bmatrix} \quad (22.177)$$

and

$$\vec{p} = \begin{bmatrix} x \\ y \\ z + \|\vec{r}\|_0 \end{bmatrix} \quad (22.178)$$

where ϕ is the latitude of the reference point of the ENU frame \vec{r}_0 is the vector from the Earth center to the origin of the ENU reference frame, i.e. the sum of the Earth mean radius and origin altitude.

Likewise, for the ellipsoidal-Earth gravity model and ENU reference frame, one has

$$\vec{a} = \vec{a}_{G,e-E}(\vec{p}) - 2[\vec{\omega}_E]_{\times} \vec{v} - [\vec{\omega}_E]_{\times}^2 \vec{p} \quad (22.179)$$

For these coast models, one may employ the EKF as the *de facto* standard for tracking. However, the linearization error needed for covariance time update may not be sufficiently accurate for relatively long time intervals. However, for high sampling rates, i.e. Δt small, one may use a simple piecewise-constant acceleration model across time steps, $k \rightarrow k + 1$, i.e.

$$\vec{a}(\hat{\vec{x}}_{k|k}) = \hat{\vec{a}}_{k|k} = \begin{bmatrix} \hat{\ddot{x}}_{k|k} \\ \hat{\ddot{y}}_{k|k} \\ \hat{\ddot{z}}_{k|k} \end{bmatrix} \quad (22.180)$$

then, for the discrete-time state equation, one has for the x -coordinate

$$\begin{bmatrix} x_{k+1} \\ \dot{x}_{k+1} \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ \dot{x}_k \end{bmatrix} + \begin{bmatrix} \frac{1}{2} \Delta t^2 \\ \Delta t \end{bmatrix} (\hat{\vec{x}}_{k|k} + w_x) \quad (22.181)$$

with w_x as some small zero-mean white-noise with variance q_x^2 and likewise for the y - and z -coordinates. In practice, the values of q_x , q_y and q_z are design parameters.

Reentry Motion Models

In reentry motion models, one typically considers gravity and aerodynamics as the significant forces. Thus, one has for the total acceleration

$$\dot{\vec{x}} = \begin{bmatrix} \vec{v} \\ \vec{a}_G + \vec{a}_A \end{bmatrix} \quad (22.182)$$

However, one may consider two types of targets: non-maneuvering and maneuvering where non-maneuvering will only have zero-lift drag while maneuvering may also have lift and lift-induced drag.

For non-maneuvering reentry targets, if one has a *known* drag parameter, α_D , the total acceleration written in the ENU reference frame simply becomes

$$\vec{a} = \vec{a}_G - 2[\vec{\omega}_E]_{\times} \vec{v} - [\vec{\omega}_E]_{\times}^2 \vec{p} - \frac{1}{2}\alpha_D \rho_0 \exp\left(-\frac{h}{h_0}\right) \|\vec{v}\|_2 \vec{v} \quad (22.183)$$

and the reentry motion model can be written with a suitable gravity model as shown previously for short time intervals. However, if one has an *unknown* drag parameter or it is time-varying or too complex to model adequately, one may instead estimate it alongside the position and velocity. This is typically done either assuming a “nearly” constant drag parameter using a Wiener process model, i.e.

$$\dot{\alpha}_D(t) = w_{\alpha}(t) \quad (22.184)$$

or a first-order Markov process, i.e.

$$\dot{\alpha}_D(t) = -\frac{1}{\tau_{\alpha}}\alpha_D(t) + w_{\alpha}(t) \quad (22.185)$$

where $w_{\alpha}(t)$ is zero-mean white noise with variance σ_{α}^2 as a design parameter.

Alternatively, one may use a nominal value $\bar{\alpha}$ and model this as an exponential, i.e.

$$\alpha_D = \bar{\alpha}_D e^{\delta} \quad (22.186)$$

or

$$\delta = \ln\left(\frac{\alpha_D}{\bar{\alpha}_D}\right) \quad (22.187)$$

where δ can be taken as a Wiener process, i.e.

$$\dot{\delta}(t) = w_{\delta}(t) \quad (22.188)$$

or a first-order Markov process, i.e.

$$\dot{\delta}(t) = -\frac{1}{\tau_{\delta}}\delta(t) + w_{\delta}(t) \quad (22.189)$$

where $w_{\delta}(t)$ is zero-mean white noise with variance σ_{δ}^2 as a design parameter.

For maneuvering reentry targets, one must account for possible lift and lift-induced drag. As such, one must choose the two coordinates for which one can decompose the lift. A common reference frame is the **velocity-turn-climb (VTC) reference frame** which defines the turn direction as parallel to the LTP and

perpendicular to the velocity and the climb direction as perpendicular to both. Thus, defining the unit vectors for the turn and climb directions as \vec{u}_t and \vec{u}_c , respectively, one has

$$\vec{u}_t = \vec{u}_h \times \vec{u}_v \quad \vec{u}_c = \vec{u}_v \times \vec{u}_t \quad (22.190)$$

where \vec{u}_h is the normal to the LTP, e.g. $[0\ 0\ 1]^T$ in ENU. Then, the acceleration due to aerodynamic forces becomes

$$\vec{\alpha}_A = \frac{1}{2} \rho_0 \exp\left(-\frac{h}{h_0}\right) \|\vec{v}\|_2^2 \begin{bmatrix} \vec{u}_v & \vec{u}_t & \vec{u}_c \end{bmatrix} \vec{\alpha}_A \quad (22.191)$$

where the **aerodynamic parameter vector**, $\vec{\alpha}_A$ can be written as

$$\vec{\alpha}_A = \begin{bmatrix} -(1 + \lambda_0)\alpha_D \\ \alpha_t \\ \alpha_c \end{bmatrix} \quad (22.192)$$

and λ_0 is the ratio of the lift to the **critical lift**, i.e. the lift at the maximum lift-to-drag ratio.

Here, $\alpha_t > 0$ corresponds to a left turning maneuver, $\alpha_t < 0$ corresponds to a right turning maneuver, $\alpha_c > 0$ corresponds to a climbing maneuver, and $\alpha_c < 0$ corresponds to a diving maneuver. It should also be pointed out that $\vec{\alpha}_A$ corresponds to four generally unknown parameters which will not generally be observable in an estimator as one cannot distinguish between λ_0 and α_D . Thus, one typically uses

$$\vec{\alpha}_A = \begin{bmatrix} -\tilde{\alpha}_D \\ \alpha_t \\ \alpha_c \end{bmatrix} \quad (22.193)$$

For the total acceleration, one must transform this to the ENU reference frame through a rotation matrix, $C_{ENU \leftarrow VTC}$, i.e.

$$C_{ENU \leftarrow VTC} = \begin{bmatrix} \cos \epsilon \cos \psi & -\cos \psi & \sin \epsilon \cos \psi \\ \cos \epsilon \sin \psi & \sin \psi & \sin \epsilon \sin \psi \\ -\sin \epsilon & 0 & \cos \epsilon \end{bmatrix} \quad (22.194)$$

where the heading angle is

$$\psi = \tan^{-1} \frac{\dot{y}}{\dot{x}} \quad (22.195)$$

and the negative elevation angle is

$$\epsilon = \tan^{-1} \frac{-\dot{z}}{\sqrt{\dot{x}^2 + \dot{y}^2}} \quad (22.196)$$

For the motion model, one must choose either the ENU or ECI frames for the gravitational force where the former will also require the Coriolis and centrifugal accelerations. As before, one either assumes either a scalar Wiener process or Markov process for each drag parameter in the *unknown* $\vec{\alpha}_A$.

Boost Motion Models

For boost motion models, one typically considers gravity and the net acceleration, i.e. thrust minus drag, as the significant forces. Thus, one has for the total acceleration

$$\dot{\vec{x}} = \begin{bmatrix} \vec{v} \\ \vec{a}_G + \vec{a}_{T-D} \end{bmatrix} \quad (22.197)$$

Here, the most general motion model uses one of the gravity models and either a Wiener process or Markov process for the net acceleration, \vec{a}_{T-D} .

However, to take advantage of the kinematics of a typical boost phase, one may use a **gravity turn (GT) motion model** which is characterized by the thrust being parallel to the Earth-relative velocity vector. In the ECEF reference frame, the GT constraint implies that non-gravitational acceleration are related through a **acceleration-to-speed ratio (ASR)**, λ_{ASR} , i.e.

$$\vec{a} = \lambda_{ASR} \vec{v} \quad (22.198)$$

Thus, a simple GT motion model can be written in state-space form in continuous-time as

$$\begin{bmatrix} \vec{v} \\ \vec{a} \end{bmatrix} = \begin{bmatrix} 0_{3 \times 3} & I_{3 \times 3} \\ 0_{3 \times 3} & \lambda_{ASR} I_{3 \times 3} \end{bmatrix} \begin{bmatrix} \vec{p} \\ \vec{v} \end{bmatrix} \quad (22.199)$$

and in discrete-time as

$$\begin{bmatrix} \vec{p}_{k+1} \\ \vec{v}_{k+1} \end{bmatrix} = \begin{bmatrix} I_{3 \times 3} & \frac{1}{\lambda_{ASR,k}} (\exp(\lambda_{ASR,k} \Delta t) - 1) I_{3 \times 3} \\ 0_{3 \times 3} & \exp(\lambda_{ASR} \Delta t) I_{3 \times 3} \end{bmatrix} \begin{bmatrix} \vec{p}_k \\ \vec{v}_k \end{bmatrix} \quad (22.200)$$

which is noticeably very similar to the CT models where λ_{ASR} may be known *a priori*, adaptively estimated, or modeled using a Wiener or Markov process. In addition, this GT model must still be gravity-compensated in application, typically as a piecewise-constant model.

For rocket boost motion models where the mass rate has significant effects on the target dynamics, one may use a **rocket gravity turn (GT) motion model**

$$\vec{v}_r = \vec{v} - \vec{\omega} \times \vec{p} \quad (22.201)$$

or

$$\vec{a}_{T-D} = \alpha_a \frac{\vec{v}_r}{\|\vec{v}_r\|} \quad (22.202)$$

where $\alpha_a = \|\vec{a}_{T-D}\|$. Thus, the total acceleration is

$$\vec{a} = \alpha_a \frac{\vec{v}_r}{\|\vec{v}_r\|} + \vec{a}_G \quad (22.203)$$

and since the drag is in the opposite direction of the velocity, one has

$$\alpha_a = \frac{T - D}{m} \quad (22.204)$$

which explicitly models GT for rockets as opposed to general GT models using the λ_{ASR} parameter.

It is often assumed that the magnitude of the non-gravitational net force, $F(t) = T(t) - D(t) = F(t_0)$ is constant where t_0 is an arbitrary initial time. Then, for $t \geq t_0$,

$$\alpha_a(t) = \frac{F(t)}{m(t)} = \frac{F(t_0)}{m(t_0) - (t - t_0)\delta} = \frac{\alpha_a(t_0)}{1 - \beta_a(t_0)(t - t_0)} \quad (22.205)$$

and

$$\beta_a(t) = \frac{\delta}{m(t)} = \frac{\delta}{m(t_0) - (t - t_0)\delta} = \frac{\beta_a(t_0)}{1 - \beta_a(t_0)(t - t_0)} \quad (22.206)$$

where $\beta_a = -\dot{m}/m$. From differentiation, one has

$$\dot{\alpha}_a = \alpha_a \beta_a \quad (22.207)$$

and

$$\dot{\beta}_a = \beta_a^2 \quad (22.208)$$

Then, the eight-state GT model for the state vector

$$\vec{x} = \begin{bmatrix} \vec{p} \\ \vec{v} \\ \alpha_a \\ \beta_a \end{bmatrix} \quad (22.209)$$

and the state rate is given by

$$\dot{\vec{x}} = \begin{bmatrix} \vec{v} \\ \alpha_a \frac{\vec{v}_r}{\|\vec{v}_r\|} + \vec{a}_G \\ \alpha_a \beta_a \\ \beta_a^2 \end{bmatrix} \quad (22.210)$$

which is a nonlinear model and can be difficult to use in filtering.

A generalization of this rocket GT model to the general boost case in VTC coordinates allows one to include the possible lateral forces during boost as velocity acceleration, α_v , turning acceleration, α_t , and climbing acceleration, α_c , as the net acceleration

$$\vec{a}_{T-D} = [\vec{u}_v \quad \vec{u}_t \quad \vec{u}_c] \begin{bmatrix} \alpha_v \\ \alpha_t \\ \alpha_c \end{bmatrix} \quad (22.211)$$

where

$$\alpha_v = \frac{T_v - D}{m} \quad (22.212)$$

$$\alpha_t = \frac{T_t + L_t}{m} \quad (22.213)$$

and

$$\alpha_c = \frac{T_c + L_c}{m} \quad (22.214)$$

where $\vec{T} = [T_v \ T_t \ T_c]^T$ is the thrust force, D is the magnitude of the drag force, L_t and L_c are the turning and climbing lift forces.

Then, the total acceleration, \vec{a} , in ENU coordinates can be written as

$$\vec{a} = \vec{a}_G + C_{ENU \leftarrow VTC} \begin{bmatrix} \alpha_v \\ \alpha_t \\ \alpha_c \end{bmatrix} \quad (22.215)$$

where

$$C_{ENU \leftarrow VTC} = [\vec{u}_v \quad \vec{u}_t \quad \vec{u}_c] \quad (22.216)$$

Then, under the same constant net force in the moving VTC reference frame, one has a ten-state model for the state vector

$$\vec{x} = \begin{bmatrix} \vec{p} \\ \vec{v} \\ \alpha_v \\ \alpha_t \\ \alpha_c \\ \beta_a \end{bmatrix} \quad (22.217)$$

with state equation

$$\dot{\vec{x}} = \begin{bmatrix} \vec{v} \\ \vec{a}_G + C_{ENU \leftarrow VTC} \begin{bmatrix} \alpha_v \\ \alpha_t \\ \alpha_c \end{bmatrix} \\ \beta_a \begin{bmatrix} \alpha_v \\ \alpha_t \\ \alpha_c \end{bmatrix} \\ \beta_a^2 \end{bmatrix} \quad (22.218)$$

where it should be noted that one should add zero-mean white noise to this model.

Advanced Navigation & Target Tracking Systems

23.1 Multi-Object Filtering Based on Random Finite Sets

The probability hypothesis density (PHD) filter is an approximation developed to overcome the computational intractability of the multi-target Bayes filter by recursively updating the first-order statistical moment of the multi-target state, i.e. the **probability hypothesis density**, also known as the **intensity**, instead of the entire posterior multi-target density. This formulation takes advantage of the special properties of Poisson RFS. This strategy is similar to the constant-gain Kalman filter which recursively updates the first moment, i.e. the mean, of the single-target state.

Optimal Multi-Target Bayes Filter

By analogy with the single-target filtering problem, the multi-target filtering problem should use two models, one for prediction of the multi-target state and one for the correction of the multi-target state using the measurement set. In general, given a multi-target state \mathbf{x}_{k-1} at time step $k-1$, each $\vec{x}_{k-1} \in \mathbf{x}_{k-1}$ either continues to exist at time step k with **survival probability**, $p_{s,k}(\vec{x}_{k-1})$, or dies with probability, $1 - p_{s,k}(\vec{x}_{k-1})$. Furthermore, conditional on the existence at time step k , the transition PDF from state \vec{x}_{k-1} to \vec{x}_k is given by $f_{k|k-1}(\vec{x}_k | \vec{x}_{k-1})$. Thus, for a given state, $\vec{x}_{k-1} \in \mathbf{x}_{k-1}$ at time step $k-1$, its behavior at the k can be modeled as the Bernoulli **survival RFS** $\mathbf{S}_{k|k-1}(\vec{x}_{k-1})$ that either takes on $\{\vec{x}_k\}$ when the target survives or \emptyset when the target dies. Furthermore, a new target at time step k can appear either by spawning from another target at time $k-1$, represented by the **spawn RFS**, $\mathbf{G}_{k|k-1}(\vec{x}_{k-1})$, or a birth independent of any existing target, represented by the **birth RFS**, \mathbf{B}_k .

Thus, given a multi-target state \mathbf{x}_{k-1} at time $k-1$, the multi-target state \mathbf{x}_k at time step k is given by the union of the surviving targets, the spawned targets, and the birthed targets, i.e. the **multi-target process equation**

$$\mathbf{X}_k = \left[\bigcup_{\vec{x}_k \in \mathbf{x}_{k-1}} \mathbf{S}_{k|k-1}(\vec{x}_k) \right] \cup \mathbf{B}_k \cup \left[\bigcup_{\vec{x}_k \in \mathbf{x}_{k-1}} \mathbf{G}_{k|k-1}(\vec{x}_k) \right] \quad (23.1)$$

which can be related to a Markov **multi-target transition density**

$$f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) \quad (23.2)$$

which captures the underlying models of target motion, survival, spawns and births.

For the measurement model, let $\mathcal{Y} = \mathbb{R}^{n_y}$ denote the single-measurement space whose elements are vectors of the form $\vec{y} = [y_1 \dots y_{n_y}]^T$ where y_1, \dots, y_{n_y} in the set of real numbers, e.g. pseudorange, pseudoazimuth. Thus, the measurement set of a multi-target system is a finite set of measurement vectors, $\mathbf{y} = \{\vec{y}_1, \dots, \vec{y}_M\}$ where M is the random number of measurements and $\vec{y}_1, \dots, \vec{y}_M$ are the individual measurements. The multi-target measurement space is the class of finite subsets of \mathcal{Y} , i.e. $\mathcal{F}(\mathcal{Y})$. This allows one to define the random measurement set as RFS, \mathbf{Y} . The multi-target measurement \mathbf{y}_k at time step k is given by the union of the detected target-originating measurements, represented by the Bernoulli **detection RFS**, $\mathbf{X}_k(\mathbf{x}_k)$, and the clutter-originating measurements, represented by the **clutter RFS**, \mathbf{C}_k , i.e. the **multi-target measurement equation**

$$\mathbf{Y}_k = \left[\bigcup_{\vec{x}_k \in \mathbf{x}_k} \mathbf{D}_k(\mathbf{x}_k) \right] \cup \mathbf{C}_k \quad (23.3)$$

which can be related to a **multi-target likelihood function**

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_k|\mathbf{x}_k) \quad (23.4)$$

It should be noted that each of the RFSs here are assumed independent of each other as in other Bayes formulations. Lastly, the actual forms of $\mathbf{G}_{k|k-1}$, \mathbf{B} , and \mathbf{C}_k are problem dependent.

Thus, all information about the multi-target state history through time step k is encapsulated in the **multi-target posterior density**, $f_{\mathbf{X}_{0:k}}(\mathbf{X}|\mathbf{y}_{1:k})$ which can be recursively computed from an initial *a priori* density, via the multi-target Bayes recursion, i.e.

$$f_{\mathbf{X}_{0:k}}(\mathbf{x}|\mathbf{y}_{1:k}) \propto f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_k|\mathbf{x}_k) f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) f_{\mathbf{X}_{0:k-1}}(\mathbf{x}_{0:k-1}|\mathbf{y}_{1:k}) \quad (23.5)$$

where individual target trajectories can be accommodated by incorporating a label in each target's state vector. For online multi-target tracking, the **multi-target filtering density** can be computed using the Bayes prediction and correction equations, i.e.

$$f_{\mathbf{X}_{k|k-1}}(\mathbf{x}_k|\mathbf{y}_{1:k-1}) = \int f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) f_{\mathbf{X}}(\mathbf{x}_{k-1}|\mathbf{y}_{1:k-1}) \delta \mathbf{x}_{k-1}$$

$$f_{\mathbf{X}}(\mathbf{x}_k|\mathbf{y}_{1:k}) = \frac{f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_k|\mathbf{x}_k) f_{\mathbf{X}_{k|k-1}}(\mathbf{x}_k|\mathbf{y}_{1:k-1})}{\int f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_k|\mathbf{x}_k) f_{\mathbf{X}_{k|k-1}}(\mathbf{x}_k|\mathbf{y}_{1:k-1}) \delta \mathbf{x}_k} \quad (23.6)$$

which are the RFS-defined Chapman-Kolmogorov equation and Bayes rule. However, these general equations do not provide an analytical form, particle filters based on the multi-target Bayes recursions have been formulated; albeit, typically only for a few number of target tracks. Four other analytical approximating filters have been developed under this framework, namely, the PHD, the cardinalized PHD, and the generalized labeled multi-Bernoulli filters. Each of these will be discussed in this textbook.

In estimation, one must use RFS models and convert into their respective density functions. That is, one requires a systematic procedure for constructing the true multi-target measurement density $f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}|\mathbf{x})$ that

faithfully describes the multi-sensor/multi-target measurement model, i.e. the likelihood that a multi-sensor measurement finite set, \mathbf{y} will be collected if targets with finite set \mathbf{x} are present. Likewise, one requires a general, systematic procedure for constructing the true multi-target Markov density $f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1})$ which faithfully describes the multi-target motion model, i.e. the likelihood that the targets will have finite set \mathbf{X}_k at time step k if they had finite set \mathbf{X}_{k-1} at time step $k - 1$. However, for the construction of true multi-target measurement and Markov PDFs, one requires the inverse operation of set integrals, i.e. **set derivatives**, which, in turn requires an understanding of belief mass functions. These set integrals, set derivatives, BMFs, PGFLs, and PGFL derivatives are all used to prove the optimality of filters used in RFS-based tracking where it should be pointed out that the set and PGFL derivatives can be computed using rules similar to those of calculus, e.g. chain, product, sum, constant, power rules.

Thus, given the measurement and motion models previously, one can first construct their belief mass functions as

$$\beta_k(\mathcal{S}|\mathbf{x}) = \Pr(\mathbf{y}_k \subseteq \mathcal{S}) = \int_{\mathcal{S}} f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_k|\mathbf{x}_k) \delta \mathbf{y}_k \quad (23.7)$$

and

$$\beta_{k|k-1}(\mathcal{S}|\mathbf{x}) = \Pr(\mathbf{x}_{k|k-1} \subseteq \mathcal{S}) = \int_{\mathcal{S}} f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) \delta \mathbf{x}_k \quad (23.8)$$

From these definitions, one can show

$$f_{\mathbf{Y}|\mathbf{X}}(\mathbf{y}_k|\mathbf{x}_k) = \frac{\delta \beta_k}{\delta \mathbf{y}}(\emptyset|\mathbf{x}) \quad (23.9)$$

and

$$f_{k|k-1}(\mathbf{x}_k|\mathbf{x}_{k-1}) = \frac{\delta^N \beta_{k|k-1}}{\delta \mathbf{x}_k}(\emptyset|\mathbf{x}_{k-1}) \quad (23.10)$$

where for $F(\mathbf{x})$ of finite set \mathbf{X} , its **set derivatives** are defined as

$$\frac{\delta F}{\delta \vec{x}}(\mathcal{S}) = \lim_{V(\mathcal{E}_x) \rightarrow 0} \frac{F(\mathcal{S} \cup \mathcal{E}_x) - F(\mathcal{S})}{V(\mathcal{E}_x)} \quad (23.11)$$

and

$$\frac{\delta \beta_{k|k-1}}{\delta \mathbf{x}_k}(\mathcal{S}) = \frac{\delta^N \beta}{\delta \vec{x}_N \cdots \delta \vec{x}_1}(\mathcal{S}) = \frac{\delta}{\delta \vec{x}_N} \frac{\delta^{N-1} \beta}{\delta \vec{x}_{N-1} \cdots \delta \vec{x}_1}(\mathcal{S}) \quad (23.12)$$

where \mathcal{E}_x is a small neighborhood of \vec{x} and $V(\mathcal{S})$ is the hyper-volume of set \mathcal{S} . Likewise, the multi-target posterior PDFs are

$$f_{\mathbf{X}|\mathbf{y}_{1:k}}(\mathbf{x}_{1:k}) = \frac{\delta \beta_{k|k}}{\delta \mathbf{x}}(\emptyset|\mathbf{y}_{1:k}) \quad (23.13)$$

as constructed by their belief mass functions

$$\beta_{k|k}(\mathcal{S}|\mathbf{y}_{1:k}) = \Pr(\mathbf{x}_{k|k} \subseteq \mathcal{S}) = \int_{\mathcal{S}} f_{\mathbf{X}}(\mathbf{x}_k|\mathbf{y}_{1:k}) \delta \mathbf{x} \quad (23.14)$$

These general models allow for non-homogeneous, non-Poisson clutter, and state-dependent detection probabilities in a principled way, depending on the optimal Bayes filter or approximations to the optimal which often use these set derivatives to prove their approximations.

23.2 Probability Hypothesis Density Filtering

Thus, based on these facts for Poisson RFSs, the PHD filter relies on three assumptions.

- A. Each target evolves and generates measurements independently of one another.
- B. The clutter RFS, \mathbf{C}_k , is Poisson with PHD $\kappa_k(\vec{y})$ and is independent of the target-originated measurements.
- C. The prior multi-target RFS governed by $f_{\vec{X}_k|\vec{X}_{k-1}}(\mathbf{x}|\mathbf{z})$ is Poisson.

It can be shown that assumption C. is completely satisfied if the birth RFS, \mathbf{B}_k , is Poisson with PHD $\beta_k(\vec{x})$ and there is no spawn RFS, though one can still include $\mathbf{G}_k(\vec{x}|\vec{z})$ with PHD $\gamma_{k|k-1}(\vec{x}|\vec{z})$ and approximate the posterior density as Poisson.

Next, let $v_k(\vec{x})$ denote the PHD associated with the multi-target posterior density, $f_{\mathbf{x}}(\mathbf{x}_k|\mathbf{y}_{1:k})$, and let $v_{k|k-1}(\vec{x})$ denote the PHD associated with the multi-target prior density, $f_{\mathbf{X}_k|\mathbf{X}_{k-1}}(\mathbf{x}_k|\mathbf{y}_{1:k-1})$. Then, with assumptions A.-C., the prior and posterior PHDs can be updated with the **probability hypothesis density filter** using the recursive prediction and correction steps

$$\begin{aligned} v_{k|k-1}(\vec{x}) &= \int p_{S,k}(\vec{z}) f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}|\vec{z}) v_{k-1}(\vec{z}) d\vec{z} \\ &\quad + \int \gamma_{k|k-1}(\vec{x}|\vec{z}) v_{k-1}(\vec{z}) d\vec{z} + \beta_k(\vec{x}) \\ v_k(\vec{x}) &= (1 - p_{D,k}) v_{k|k-1}(\vec{x}) \\ &\quad + \sum_{\vec{y} \in \mathbf{y}_k} \frac{p_{D,k}(\vec{x}) f_{\vec{Y}_k|\vec{X}_k}(\vec{y}|\vec{x}) v_{k|k-1}(\vec{x})}{\kappa_k(\vec{y}) + \int p_{D,k}(\vec{z}) f_{\vec{Y}_k|\vec{X}_k}(\vec{y}|\vec{z}) v_{k|k-1}(\vec{z}) d\vec{z}} \end{aligned} \quad (23.15)$$

where \vec{z} is a dummy integration vector.

It is important to note that the PHD filter completely avoids the combinatorial computations arising from the unknown track-to-measurement. Furthermore, since the posterior PHD is a function on the single-target state space, \mathcal{X} , the PHD filter requires much less computational power than the multi-target Bayes filter. However, the PHD filter does not admit a closed-form solution, in general, and **particle-PHD** approximations suffer from the “curse of dimensionality” of the number of targets and particles required to sufficiently cover the single-target space. Notably, these also require some form of clustering algorithm in order to extract meaningful tracks. However, a closed-form solution exists for linear, Gaussian models for individual targets as will be shown in the following subsection.

Gaussian Mixture-Probability Hypothesis Density Filter

A closed-form to the PHD filter can be obtained by assuming the following

- D. Each target and sensor follows a linear-Gaussian dynamical model, i.e.

$$\begin{aligned} f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}|\vec{z}) &= f_N(\vec{x}; F_{k-1}\vec{z}, Q_{k-1}) \\ f_{\vec{Y}_k|\vec{X}_k}(\vec{y}|\vec{z}) &= f_N(\vec{y}; H_k\vec{z}, R_k) \end{aligned} \quad (23.16)$$

E. The survival and detection probabilities are state independent, i.e.

$$\begin{aligned} p_{S,k}(\vec{x}) &= p_{S,k} \\ p_{D,k}(\vec{x}) &= p_{D,k} \end{aligned} \quad (23.17)$$

F. The birth and spawn RFS have Gaussian mixture (GM) PHDs, i.e.

$$\begin{aligned} \beta_k(\vec{x}) &= \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(j)} f_N(\vec{x}; \vec{m}_{\beta,k}^{(j)}, P_{\beta,k}^{(j)}) \\ \gamma_{k|k-1}(\vec{x}|\vec{z}) &= \sum_{j=1}^{J_{\beta,k}} w_{\gamma,k}^{(j)} f_N(\vec{x}; F_{\gamma,k-1}^{(j)} \vec{z} + \vec{d}_{\gamma,k}^{(j)}, Q_{\gamma,k-1}^{(j)}) \end{aligned} \quad (23.18)$$

where $J_{\beta,k}$ is the number of terms in the birth GM-PHD, $w_{\beta,k}^{(j)}$ is the weight of the j^{th} term in the birth GM-PHD, $\vec{m}_{\beta,k}^{(j)}$ is the mean of the j^{th} term in the birth GM-PHD, $P_{\beta,k}^{(j)}$ is the covariance of the j^{th} term in the birth GM-PHD, $J_{\gamma,k}$ is the number of terms in the spawn GM-PHD, $w_{\gamma,k}^{(j)}$ is the weight of the j^{th} term in the spawn GM-PHD, $F_{\gamma,k-1}^{(j)} \vec{z} + \vec{d}_{\gamma,k}^{(j)}$ is the mean of the j^{th} term in the spawn GM-PHD, and $Q_{\gamma,k-1}^{(j)}$ is the covariance of the j^{th} term in the spawn GM-PHD.

In this model, it should be noted that the mean of the birth PHD, $\vec{m}_{\beta,k}$ corresponds to the highest local concentrations of expected number of spontaneous births, whereas the mean of the spawn PHD, $F_{\gamma,k-1}^{(j)} \vec{z}_{k-1} + \vec{d}_{\gamma,k}^{(j)}$ corresponds to the highest local concentration of the expected number of generated spawns which depends on the previous posterior state. For example, $\vec{m}_{\beta,k}$ could be a static, known, airport location while \vec{z}_{k-1} could be an aircraft carrier which is also being tracked, but can spawn fighter planes. For both, $w_{\bullet,k}^{(j)}$ corresponds to the expected number of targets originating from any particular Gaussian term, and $P_{\beta,k}^{(j)}$ and $Q_{\gamma,k-1}^{(j)}$ corresponds to the spread of the PHDs in the vicinity of the peaks.

However, it should be noted that one can derive closed-form expressions for exponential mixtures for $p_{S,k}(\vec{x})$ and $p_{D,k}(\vec{x})$, i.e.

$$\begin{aligned} p_{S,k}(\vec{x}) &= w_{S,k}^{(0)} + \sum_{j=1}^{J_{S,k}} w_{S,k}^{(j)} f_N(\vec{x}; \vec{m}_{S,k}^{(j)}, P_{S,k}^{(j)}) \\ p_{D,k}(\vec{x}) &= w_{D,k}^{(0)} + \sum_{j=1}^{J_{D,k}} w_{D,k}^{(j)} f_N(\vec{x}; \vec{m}_{D,k}^{(j)}, P_{D,k}^{(j)}) \end{aligned} \quad (23.19)$$

Furthermore, the use of Gaussian mixture PHDs allows one to approximate any other form of birth and spawn PHDs to any desired accuracy, albeit at a higher memory and computational cost due to the number of mixtures required to do so.

With assumptions A.-F. and an initial multi-target state as a Gaussian mixture, then, the **Gaussian mixture - probability hypothesis density (GM-PHD) filter** can be constructed as recursive prediction and correction steps. For the prediction step, the prior PHD at time step k will be a Gaussian mixture

$$\nu_{k|k-1}(\vec{x}) = \nu_{S,k|k-1}(\vec{x}) + \nu_{\beta,k}(\vec{x}) + \nu_{\gamma,k|k-1}(\vec{x}) \quad (23.20)$$

where the posterior PHD at time step $k-1$ is a Gaussian mixture

$$\nu_{k-1}(\vec{x}) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} f_N \left(\vec{x}; \vec{m}_{k-1}^{(i)}, P_{k-1}^{(i)} \right) \quad (23.21)$$

and the individual PHDs are Gaussian mixtures

$$\nu_{S,k|k-1}(\vec{x}) = p_{S,k} \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} f_N \left(\vec{x}; F_{k-1} \vec{m}_{k-1}^{(i)}, F_{k-1} P_{k-1}^{(i)} F_{k-1}^T + Q_{k-1} \right) \quad (23.22)$$

$$\nu_{\beta,k}(\vec{x}) = \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(j)} f_N \left(\vec{x}; \vec{m}_{\beta,k}^{(j)}, P_{\beta,k}^{(j)} \right) \quad (23.23)$$

$$\nu_{\gamma,k|k-1}(\vec{x}) = \sum_{i=1}^{J_{k-1}} \sum_{j=1}^{J_{\gamma,k}} w_{k-1}^{(i)} w_{\gamma,k}^{(j)} f_N \left(\vec{x}; F_{\gamma,k-1} \vec{m}_{k-1}^{(i)} + \vec{d}_{\gamma,k}^{(j)}, F_{\gamma,k-1} P_{k-1}^{(i)} \left(F_{\gamma,k-1}^{(j)} \right)^T + Q_{\gamma,k-1}^{(j)} \right) \quad (23.24)$$

Likewise, for the correction step, the posterior multi-target PHD at time step k will be a Gaussian mixture

$$\nu_k(\vec{x}) = (1 - p_{D,k}) \nu_{k|k-1}(\vec{x}) + \sum_{\vec{y} \in \mathbf{y}_k} \sum_{i=1}^{J_{k|k-1}} w_k^{(i)}(\vec{y}) f_N \left(\vec{x}; \vec{m}_{k|k}^{(i)}(\vec{y}), P_{k|k}^{(i)}(\vec{y}) \right) \quad (23.25)$$

where the prior multi-target state PHD at time step k is a Gaussian mixture

$$\nu_{k|k-1}(\vec{x}) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} f_N \left(\vec{x}; \vec{m}_{k|k-1}^{(i)}, P_{k|k-1}^{(i)} \right) \quad (23.26)$$

and the Gaussian mixture terms for each measurement, \vec{y} , are given by

$$w_k^{(i)}(\vec{y}) = \frac{\mathcal{L}_k^{(i)}(\vec{y})}{\kappa_k(\vec{y}) + \sum_{j=1}^{J_{k|k-1}} \mathcal{L}_k^{(j)}(\vec{y})} \quad (23.27)$$

$$\mathcal{L}_k^{(i)}(\vec{y}) = p_{D,k} w_{k|k-1}^{(i)} f_N(\vec{y}; H_k \vec{m}_{k|k-1}^{(i)}, S_k^{(i)}) \quad (23.28)$$

$$S_k^{(i)} = H_k P_{k|k-1}^{(i)} H_k^T + R_k \quad (23.29)$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} H_k^T \left(S_k^{(i)} \right)^{-1} \quad (23.30)$$

$$\vec{m}_{k|k}^{(i)}(\vec{y}) = \vec{m}_{k|k-1}^{(i)} + K_k^{(i)} (\vec{y} - H_k \vec{m}_{k|k-1}^{(i)}) \quad (23.31)$$

and

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} S_k^{(i)} \left(K_k^{(i)} \right)^T \quad (23.32)$$

where it should be pointed out that the mean and covariance recursions for the surviving and detected targets are given by the classic Kalman filter equations. Thus, for nonlinear motion and measurement models, one could easily implement an extended or unscented Kalman filter-like versions of the GM-PHD filter through either Jacobian linearization or statistical linearization.

Furthermore, based on the properties of Poisson PHDs, the prior cardinality estimate can be shown to be

$$\hat{N}_{k|k-1} = \hat{N}_{k-1} \left(p_{S,k} + \sum_{j=1}^{J_{\gamma,k}} w_{\gamma,k}^{(i)} \right) + \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(i)} \quad (23.33)$$

and the posterior cardinality estimate can be shown to be

$$\hat{N}_k = \hat{N}_{k|k-1} (1 - p_{D,k}) + \sum_{\vec{y} \in \mathbf{y}_k} \sum_{i=1}^{J_{k|k-1}} w_k^{(i)}(\vec{y}) \quad (23.34)$$

Thus, the prior multi-target PHD consists of the sum of the survival, birth, and spawn GMs while the posterior multi-target PHD consists of the sum of each of these GMs possibly being missed detected as well as detected *for each measurement*. Thus, the number of terms at time step k , J_k , is

$$J_k = (J_{k-1}(1 + J_{\gamma,k}) + J_{\beta,k})(1 + |\mathbf{y}_k|) \quad (23.35)$$

which increases without bound. This combinatorial nature of the GM-PHD requires that practical implementations of a GM-PHD filter to implement an additional **GM-PHD manage step** made up of pruning very unlikely terms based on the individual weight thresholds, merging similar terms based on the Mahalanobis distance, and capping the maximum number of terms in the posterior multi-target PHD based on the individual weight thresholds.

After pruning and before capping, the merging step evaluates the set of mixture terms with indices I , as

$$j = \underset{i \in I}{\operatorname{argmax}} w_k^{(i)} \quad (23.36)$$

and computes a subset of indices, L , for Gaussian terms ($\vec{w}_k^{(i)}$, $\vec{m}_k^{(i)}$, $\vec{P}_k^{(i)}$) that are statistically close to this term based on thresholding the Mahalanobis distance, i.e.

$$L = \{i \in I : (\vec{m}_k^{(i)} - \vec{m}_k^{(j)})^T \left(P_k^{(i)} \right)^{-1} (\vec{m}_k^{(i)} - \vec{m}_k^{(j)}) \leq \tau\} \quad (23.37)$$

for some chosen τ . This allows one to form the new merged Gaussian term ($\bar{w}_k^{(\ell)}$, $\bar{m}_k^{(\ell)}$, $\bar{P}_k^{(\ell)}$), with index ℓ as

$$\bar{w}_k^{(\ell)} = \sum_{i \in L} w_k^{(i)} \quad (23.38)$$

$$\bar{m}_k^{(\ell)} = \frac{1}{\bar{w}_k^{(\ell)}} \sum_{i \in L} w_k^{(i)} \vec{m}_k^{(i)} \quad (23.39)$$

and

$$\bar{P}_k^{(\ell)} = \frac{1}{\bar{w}_k^\ell} \sum_{i \in L} w_k^{(i)} \left(P_k^{(i)} + (\bar{m}_k^{(\ell)} - \bar{m}_k^{(i)}) (\bar{m}_k^{(\ell)} - \bar{m}_k^{(i)})^T \right) \quad (23.40)$$

and reiterates over terms with indices in I with all L removed.

Furthermore, one must also form a multi-target state estimate using an **GM-PHD extraction step**. For the particle-PHD filter, one typically performs clustering of the particles representing the PHD based on the estimated number of targets or close to that number. In the GM-PHD, one can easily extract the expected number of targets *from each term* in the mixture. Thus, if there is significant separation for each term, a condition easily satisfied due to the merging step in the manage step, then, the mean of each term with a statistically significant weight, e.g. > 0.5 , should be output as a likely target location. In addition, as a single Gaussian can represent any number of targets in its weight, one typically rounds the weight and reports that many copies of the mean.

23.3 Cardinalized Probability Hypothesis Density Filtering

The PHD filter updates the cardinality information indirectly through the PHD of the posterior multi-target density which can be integrated over the entire single-target state space to obtain the mean of the cardinality distribution. This effectively approximates the cardinality distribution by a Poisson distribution. However, as the mean and variance of a Poisson distribution are equal, when a high number of targets are present, the PHD filter estimates the cardinality with a correspondingly high variance. In practice, this approximation manifests itself in drastically fluctuating estimates of the number of targets. To address this problem, Mahler relaxed the first-order assumption on the number of targets and derived a generalization of the PHD recursion known as the cardinalized PHD (CPHD) recursion, which jointly propagates the PHD and an arbitrary cardinality distribution through the use of IID cluster RFSs. Likewise, a closed-form solution for the CPHD recursion can be written for linear, Gaussian motion and measurement models which uses a Gaussian mixture formulation called the GM-CPHD.

This section will adopt the following notation. C_j^ℓ is the **binomial coefficient**

$$C_j^\ell = \frac{\ell!}{j!(\ell-j)!} \quad (23.41)$$

\mathcal{P}_j^N is the **permutation coefficient**

$$\mathcal{P}_j^N = \frac{N!}{(N-j)!} \quad (23.42)$$

$\langle \cdot, \cdot \rangle$ denotes the **inner product** between two real-valued functions, $\alpha(\vec{x})$ and $\beta(\vec{x})$, as

$$\langle \alpha, \beta \rangle = \int \alpha(\vec{x}) \beta(\vec{x}) d\vec{x} \quad (23.43)$$

or real-valued sequences, $\alpha(\ell)$ and $\beta(\ell)$, as

$$\langle \alpha, \beta \rangle = \sum_{\ell=0}^{\infty} \alpha(\ell) \beta(\ell) \quad (23.44)$$

CPHD Filter

The CPHD filter relies on four assumptions.

- A. Each target evolves and generates measurements independently of one another.
- B. The clutter RFS, \mathbf{C}_k , is IID cluster and is independent of the target-originated measurements.
- C. The birth RFS, \mathbf{B}_k , and surviving RFS, $\mathbf{S}_k(\vec{x})$ are IID cluster and is independent of each other.
- D. The prior and posterior multi-target RFS's are IID cluster.

For the sake of simplicity of presentation, this version will also assume (E.) there is no spawn RFS, though the CPHD has been derived for Bernoulli, Poisson, and zero-inflated Poisson RFS spawn models.

Next, let $v_k(\vec{x})$ denote the PHD and $\varrho_k(N)$ denote the cardinality distribution associated with the multi-target posterior density, $f_{\mathbf{X}_k}(\mathbf{x}_k|\mathbf{y}_{1:k})$. Also, let $v_{k|k-1}(\vec{z})$ denote the PHD and $\varrho_{k|k-1}(N)$ denote the cardinality distribution associated with the multi-target prior density, $f_{\mathbf{X}_k|\mathbf{X}_{k-1}}(\mathbf{x}_k|\mathbf{y}_{1:k-1})$. Then, with assumptions A.-E., the prior and posterior PHDs and cardinality distributions can be updated with the **cardinalized probability hypothesis density (CPHD) filter** using the following recursive prediction and correction steps.

The prediction step of the CPHD filter is given by

$$\begin{aligned} v_{k|k-1}(\vec{x}) &= \int p_{S,k}(\vec{z}) f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}|\vec{z}) v_{k-1}(\vec{z}) d\vec{z} + \beta_k(\vec{x}) \\ \varrho_{k|k-1}(N) &= \sum_{j=0}^N \varrho_{B,k}(N-j) \sum_{\ell=j}^{\infty} C_j^{\ell} \frac{\langle p_{S,k}, v_{k-1} \rangle^j \langle 1 - p_{S,k}, v_{k-1} \rangle^{\ell-j}}{\langle 1, v_{k-1} \rangle^{\ell}} \varrho_{k-1}(\ell) \end{aligned} \quad (23.45)$$

where $p_{S,k}(\vec{z})$ is the probability of target survival at time k given previous state \vec{z} , \vec{z} is a dummy integration vector, $f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}|\vec{z})$ is the single-target transition density at time k given previous state \vec{z} , and $\beta_k(\cdot)$ and $\varrho_{B,k}(\cdot)$ are the birth PHD and cardinality distribution at time k , respectively. Notably, the prior PHD prediction is the same as for the PHD filter and uncoupled from the prior cardinality. The CPHD prior cardinality distribution is simply a convolution of the birth and surviving cardinalities as the prior cardinality random variable is the sum of the birth and surviving targets.

The correction step of the CPHD filter is given by

$$\begin{aligned} v_k(\vec{x}) &= \frac{\langle Y_k^1[v_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle}{\langle Y_k^0[v_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle} (1 - p_{D,k}(\vec{x})) v_{k|k-1}(\vec{x}) \\ &\quad + \sum_{\vec{y} \in \mathbf{y}_k} \frac{\langle Y_k^1[v_{k|k-1}, \mathbf{y}_k \setminus \{\vec{y}\}], \varrho_{k|k-1} \rangle}{\langle Y_k^0[v_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle} (\Psi_{k,\vec{y}}(\vec{x}) v_{k|k-1}(\vec{x})) \\ \varrho_k(N) &= \frac{\langle Y_k^0[v_{k|k-1}, \mathbf{y}_k](N), \varrho_{k|k-1}(N) \rangle}{\langle Y_k^0[v_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle} \end{aligned} \quad (23.46)$$

where

$$Y_k^u[v, \mathbf{y}](N) = \sum_{j=0}^{\min(|\mathbf{y}|, N)} (|\mathbf{y}| - j) \varrho_{C,k}(|\mathbf{y}| - j) \mathcal{P}_{j+u}^N \frac{\langle 1 - p_{D,k}, v \rangle^{N-(j+u)}}{\langle 1, v \rangle^N} e_j(\Xi_k(v, \mathbf{y})) \quad (23.47)$$

$$\Xi_k(v, \mathbf{y}) = \{\langle v, \Psi_{k, \vec{y}} \rangle : \vec{y} \in \mathbf{y}\} \quad (23.48)$$

$$\Psi_{k, \vec{y}}(\vec{x}) = \frac{\langle 1, \kappa_k \rangle}{\kappa_k(\vec{y})} f_{\vec{Y}_k | \vec{X}_k}(\vec{y} | \vec{x}) p_{D,k}(\vec{x}) \quad (23.49)$$

$$e_j(\Xi) = \sum_{\mathbf{z} \subseteq \Xi, |\mathbf{z}|=j} \left(\prod_{\vec{z} \in \mathbf{z}} \vec{z} \right) \quad (23.50)$$

is the **elementary symmetric function** of order j with $e_0(\Xi) = 1$ by convention, \mathbf{y}_k is the measurement set at time k , $p_{D,k}(\vec{x})$ is the probability of target detection at time step k , $f_{\vec{Y}_k | \vec{X}_k}(\vec{y} | \vec{x})$ is the single-target measurement likelihood at time step k given state \vec{x} , and $\kappa_k(\cdot)$ and $\varrho_{C,k}$ are the clutter PHD and cardinality distribution at time k , respectively.

The CPHD cardinality and PHD corrections are coupled; however, the posterior PHD is similar to the PHD filter as both have one missed detection term and $|\mathbf{y}|$ detection terms. The cardinality correction is a Bayes update incorporating the clutter cardinality, the measurement set, the prior PHD, and the prior cardinality into $\Upsilon_k^0[v_{k|k-1}, \mathbf{y}_k](N)$, i.e. the likelihood of the multi-target measurement \mathbf{y} where $\langle \Upsilon_k^0[v_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle$ is the normalizing constant. The particle-CPHD filter can be used to approximate this recursion for arbitrary cardinality distributions and PHDs. However, just like the particle-PHD filter, the state extraction involves clustering to partition the particle population into a given number of clusters, e.g. the estimated number of targets, but adds additional uncertainty in the target tracking system due to the potential clustering error. Furthermore, if the cardinality distribution is infinite tailed, the recursive updates of the entire posterior cardinality distribution is generally not possible as this would involve storing an infinite number of terms for the infinite sums. Thus, in practice, the cardinality distributions are truncated at and approximated with a finite number of terms which must be significantly greater than the maximum number of targets possible in the surveillance volume which must be chosen *a priori* for the CPHD filter.

Notably, the CPHD filter requires a solver of the elementary symmetric functions which can be done efficiently with the following techniques. Evaluating the elementary symmetric functions directly from the definition is clearly intractable. Using a basic result from combinatorics theory known as the Newton–Girard formulas, the elementary symmetric function $e_j(\cdot)$ can be computed using the following procedure. Let $\lambda_1, \dots, \lambda_M$ be distinct roots of the polynomial $a_M x^M + a_{M-1} x^{M-1} + \dots + a_1 x + a_0$. Then, $e_j(\cdot)$ for orders $j = 0, \dots, M$ is given by $e_j(\lambda_1, \dots, \lambda_M) = (-1)^j a_{M-j} / a_M$. Thus, the values $e_j(\mathbf{y})$ over the finite set, \mathbf{y} , can be evaluated by expanding out the polynomial with roots given by the elements of \mathbf{y} , which can be implemented using an appropriate recursion or convolution. For finite set \mathbf{y} , this calculation of $e_j(\mathbf{y})$ requires $|\mathbf{y}|^2$ operations. In the CPHD recursion, each correction step requires the calculation of $|\mathbf{y}| + 1$ elementary symmetric functions, i.e. one for \mathbf{y} and one for each set $\{\mathbf{y} \setminus \vec{y}\}$ where $\vec{y} \in \mathbf{y}$. Thus, the CPHD recursion has a complexity of $O(|\mathbf{y}|^3)$. However, in practice, the number of measurements considered in \mathbf{y} can be reduced by classical gating techniques.

Gaussian Mixture - CPHD Filter

A closed-form to the CPHD filter can be obtained by assuming the following

- F. Each target and sensor follows a linear-Gaussian dynamical model, i.e.

$$\begin{aligned} f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}|\vec{z}) &= f_N(\vec{x}; F_{k-1}\vec{z}, Q_{k-1}) \\ f_{\vec{Y}_k|\vec{X}_k}(\vec{y}|\vec{z}) &= f_N(\vec{y}; H_k\vec{z}, R_k) \end{aligned} \quad (23.51)$$

G. The survival and detection probabilities are state independent, i.e.

$$\begin{aligned} p_{S,k}(\vec{x}) &= p_{S,k} \\ p_{D,k}(\vec{x}) &= p_{D,k} \end{aligned} \quad (23.52)$$

H. The PHD of the birth RFS is a GM, i.e.

$$\beta_k(\vec{x}) = \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(j)} f_N(\vec{x}; \vec{m}_{\beta,k}^{(j)}, P_{\beta,k}^{(j)}) \quad (23.53)$$

With assumptions A.-H. and an initial multi-target state as a Gaussian mixture, then, the **Gaussian mixture - cardinalized probability hypothesis density (GM-CPHD) filter** can be constructed as recursive prediction and correction steps. For the GM-CPHD prediction step, the prior PHD and cardinality distributions at time step k are given by

$$\begin{aligned} v_{k|k-1}(\vec{x}) &= v_{S,k|k-1}(\vec{x}) + v_{\beta,k}(\vec{x}) \\ \varrho_{k|k-1}(N) &= \sum_{j=0}^N \varrho_{B,k}(N-j) \sum_{\ell=j}^{\infty} C_j^\ell \varrho_{k-1}(\ell) p_{S,k}^j (1-p_{S,k})^{\ell-j} \end{aligned} \quad (23.54)$$

where the posterior PHD at time step $k-1$ is a Gaussian mixture

$$v_{k-1}(\vec{x}) = \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} f_N(\vec{x}; \vec{m}_{k-1}^{(i)}, P_{k-1}^{(i)}) \quad (23.55)$$

the posterior cardinality distribution at time step $k-1$ is ϱ_{k-1} , and the individual PHDs are Gaussian mixtures

$$v_{S,k|k-1}(\vec{x}) = p_{S,k} \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} f_N\left(\vec{x}; F_{k-1}\vec{m}_{k-1}^{(i)}, F_{k-1}P_{k-1}^{(i)}F_{k-1}^T + Q_{k-1}\right) \quad (23.56)$$

$$v_{\beta,k}(\vec{x}) = \sum_{j=1}^{J_{\beta,k}} w_{\beta,k}^{(j)} f_N\left(\vec{x}; \vec{m}_{\beta,k}^{(j)}, P_{\beta,k}^{(j)}\right) \quad (23.57)$$

Likewise, for the GM-CPHD correction step, the posterior PHD and cardinality distributions at time step k are given by

$$\begin{aligned} v_k(\vec{x}) &= \frac{\Omega_k^1[\vec{w}_{k|k-1}, \mathbf{y}_k](N) \varrho_{k|k-1}(N)}{\langle \Omega_k^0[\vec{w}_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle} (1-p_{D,k}) v_{k|k-1}(\vec{x}) + \sum_{\vec{y} \in \mathbf{y}_k} \sum_{i=1}^{J_{k|k-1}} w_k^{(i)}(\vec{y}) f_N\left(\vec{x}; \vec{m}_{k|k}^{(i)}(\vec{y}), P_{k|k}^{(i)}(\vec{y})\right) \\ \varrho_k(N) &= \frac{\Omega_k^0[\vec{w}_{k|k-1}, \mathbf{y}_k](N) \varrho_{k|k-1}(N)}{\langle \Omega_k^0[\vec{w}_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle} \end{aligned} \quad (23.58)$$

where the prior multi-target state PHD at time step k is a Gaussian mixture

$$v_{k|k-1}(\vec{x}) = \sum_{i=1}^{J_{k|k-1}} w_{k|k-1}^{(i)} f_N(\vec{x}; \vec{m}_{k|k-1}^{(i)}, P_{k|k-1}^{(i)}) \quad (23.59)$$

$$\Omega_k^u[\vec{w}, \mathbf{y}_k](N) = \sum_{j=0}^{\min(|\mathbf{y}|, N)} (|\mathbf{y}| - j) \varrho_{C,k}(|\mathbf{y}| - j) \mathcal{P}_{j+u}^N \frac{(1 - p_{D,k})^{N-(j+u)}}{\langle 1, w \rangle^{j+u}} e_j(\Lambda_k(w, \mathbf{y})) \quad (23.60)$$

$$\Lambda_k(w, \mathbf{y}) = \left\{ \frac{\langle 1, \kappa_k \rangle}{\kappa_k(\vec{y})} p_{D,k} \vec{w}_{k|k-1}^T \vec{\mathcal{L}}_k(\vec{y}) : \vec{y} \in \mathbf{y} \right\} \quad (23.61)$$

$$\vec{w}_{k|k-1} = \left[w_{k|k-1}^{(1)}, \dots, w_{k|k-1}^{(J_{k|k-1})} \right]^T \quad (23.62)$$

$$\mathcal{L}_k^{(i)}(\vec{y}) = f_N(\vec{x}_k; H_k \vec{m}_{k|k-1}^{(i)}, S_k^{(i)}) \quad (23.63)$$

$$\vec{\mathcal{L}}_k(\vec{y}) = \left[\mathcal{L}_k^{(1)}(\vec{y}), \dots, \mathcal{L}_k^{(J_{k|k-1})}(\vec{y}) \right]^T \quad (23.64)$$

where the Gaussian mixture terms for each measurement, \vec{y} , are given by

$$w_k^{(i)}(\vec{y}) = p_{D,k} w_{k|k-1}^{(i)} \mathcal{L}_k^{(i)}(\vec{y}) \frac{\langle \Omega_k^1[\vec{w}_{k|k-1}, \mathbf{y}_k \setminus \{\vec{y}\}], \varrho_{k|k-1} \rangle \langle 1, \kappa_k \rangle}{\langle \Omega_k^0[w_{k|k-1}, \mathbf{y}_k], \varrho_{k|k-1} \rangle \kappa_k(\vec{y})} \quad (23.65)$$

$$S_k^{(i)} = H_k P_{k|k-1}^{(i)} H_k^T + R_k \quad (23.66)$$

$$K_k^{(i)} = P_{k|k-1}^{(i)} H_k^T \left(S_k^{(i)} \right)^{-1} \quad (23.67)$$

$$\vec{m}_{k|k}^{(i)}(\vec{y}) = \vec{m}_{k|k-1}^{(i)} + K_k^{(i)}(\vec{y} - H_k \vec{m}_{k|k-1}^{(i)}) \quad (23.68)$$

and

$$P_{k|k}^{(i)} = P_{k|k-1}^{(i)} - K_k^{(i)} S_k^{(i)} \left(K_k^{(i)} \right)^T \quad (23.69)$$

where it should be pointed out that the mean and covariance recursions for the surviving and detected targets are given by the classic Kalman filter equations. Thus, for nonlinear motion and measurement models, one could easily implement an extended or unscented Kalman filter-like versions of the GM-CPHD filter through either Jacobian linearization or statistical linearization. Also, similar to GM-PHD filter, the number of Gaussian components required to represent the posterior PHD increases without bound which can be mitigated using the same **manage step** as described for the GM-PHD.

For the **GM-CPHD extraction step**, the number of targets can be estimated using, for example, the expected posterior (EAP) or maximum a posteriori (MAP) estimators. However, the EAP estimator is likely to fluctuate and be unreliable as clutter and missed detections may induce minor modes in the posterior cardinality distribution, away from the target-induced primary mode which would effect the expected value. As such, the MAP estimator is typically more reliable since it ignores minor modes and locks directly onto the target-induced primary mode, thereby usually preferred over the EAP estimator.

Constant-Cardinality CPHD Filter

A special case of the CPHD filter can be derived if the number of targets, N , is known *a priori* and constant, i.e. the cardinality distribution at any time step k is a Dirac delta function at N , $\varrho_{k|k-1} = \varrho_k = \delta_N(\cdot)$. As there are no births, the birth PHD is $\beta_k(\vec{x}) = 0 \forall k$ and no deaths, the survival probability is $p_{S,k} = 1 \forall k$. Thus, the prediction and correction steps for the CPHD filter can be simplified as

$$\begin{aligned} v_{k|k-1}(\vec{x}) &= \int f_{\vec{X}_k|\vec{X}_{k-1}}(\vec{x}|\vec{z})v_{k-1}(\vec{z})d\vec{z} \\ v_k(\vec{x}) &= \frac{\Upsilon_k^1[v_{k|k-1}, \mathbf{y}_k](N)}{\Upsilon_k^0[v_{k|k-1}, \mathbf{y}_k](N)}(1 - p_{D,k}(\vec{x}))v_{k|k-1}(\vec{x}) \\ &\quad + \sum_{\vec{y} \in \mathbf{y}_k} \frac{\Upsilon_k^1[v_{k|k-1}, \mathbf{y}_k \setminus \{\vec{y}\}](N)}{\Upsilon_k^0[v_{k|k-1}, \mathbf{y}_k](N)} \Psi_{k,\vec{y}}(\vec{x})v_{k|k-1}(\vec{x}) \end{aligned} \quad (23.70)$$

Furthermore, with the assumptions F.-H., the GM-CPHD can be simplified as

$$\begin{aligned} v_{k|k-1}(\vec{x}) &= \sum_{i=1}^{J_{k-1}} w_{k-1}^{(i)} f_N \left(\vec{x}; F_{k-1} \vec{m}_{k-1}^{(i)}, F_{k-1} P_{k-1}^{(i)} F_{k-1}^T + Q_{k-1} \right) \\ v_k(\vec{x}) &= \frac{\Omega_k^1[v_{k|k-1}, \mathbf{y}_k](N)}{\Omega_k^0[v_{k|k-1}, \mathbf{y}_k](N)}(1 - p_{D,k}(\vec{x}))v_{k|k-1}(\vec{x}) \\ &\quad + \sum_{\vec{y} \in \mathbf{y}_k} \sum_{i=1}^{J_{k-1}} w_k^{(i)}(\vec{y}) f_N \left(\vec{x}; \vec{m}_{k|k}^{(i)}(\vec{y}), P_{k|k}^{(i)}(\vec{y}) \right) \end{aligned} \quad (23.71)$$

where the weights are now defined as

$$w_k^{(i)}(\vec{y}) = p_{D,k} w_{k|k-1}^{(i)} \mathcal{L}_k^{(i)}(\vec{y}) \frac{\Omega_k^1[\vec{w}_{k|k-1}, \mathbf{y}_k \setminus \{\vec{y}\}](N)}{\Omega_k^0[w_{k|k-1}, \mathbf{y}_k](N)} \frac{\langle 1, \kappa_k \rangle}{\kappa_k(\vec{y})} \quad (23.72)$$

and the other terms have the same definitions as the standard GM-CPHD. It has been shown that this CC-GM-CPHD filter has better performance than the JPDAF for resolving target crossings with lower computational cost.

23.4 Multi-Bernoulli Filtering

Conjugate Prior of Multi-Target Bayes Filter

Track-Oriented Marginalized Multi-Bernoulli Filter

Measurement-Oriented Marginalized Multi-Bernoulli Filter

References

For more information, please refer to the following

- Williams, J. L., “Marginal Multi-Bernoulli Filters: RFS Derivation of MHT, JIPDA, and Association-Based MeMBer,” *IEEE Transactions on Aerospace and Electronic Systems*, Volume: 51, Issue: 3, 2015, pp. 1664-1687

23.5 Labeled Multi-Bernoulli Filtering

Multi-Bernoulli RFS

Labeled Multi-Bernoulli Filter

Generalized Labeled Multi-Bernoulli Filter

Thus, to incorporate tracks into the multi-target Bayes filter, one can identify targets by an ordered pair of integers $\ell = (k, i)$ where k is the **time of birth** and $i \times \mathbb{N}$ is the unique index to distinguish objects born at the same time. In this way, the label space for targets born at time k is denoted $\mathbb{L}_k = \{k\} \times \mathbb{N}$ and has a single-target state $\vec{x} \in \mathbb{X} \times \mathbb{L}_k$. Thus, the label space for targets at time k , including those born prior to k , denoted as $\mathbb{L}_{0:k}$, is given recursively by the disjoint union, $\mathbb{L}_{0:k} = \mathbb{L}_{0:k-1} \cup \mathbb{L}_k$ and the multi-target state is a finite subset of $\mathbb{X} \times \mathbb{L}_{0:k}$.

Thus, for $N(k)$ targets and $M(k)$ measurements, the (labeled) multi-target state at time k is given by

$$\underline{\vec{x}} = \{\vec{x}_{k,1}, \dots, \vec{x}_{k,N(k)}\} \quad (23.73)$$

and the multi-target measurement at time k is given by

$$\mathbf{y}_k = \{\vec{y}_{k,1}, \dots, \vec{y}_{k,M}\} \quad (23.74)$$

and the optimal multi-target Bayes filter is given by the recursive prediction and correction steps as

$$\begin{aligned} f_{\underline{\mathbf{X}}|k-1}(\underline{\mathbf{x}}_k | \mathbf{y}_{1:k-1}) &= \int f_{k|k-1}(\mathbf{x}_k | \underline{\mathbf{x}}_{k-1}) f_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}_{k-1} | \mathbf{y}_{1:k-1}) \delta \underline{\mathbf{x}}_{k-1} \\ f_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}_k | \mathbf{y}_{1:k}) &= \frac{f_{\mathbf{Y}|\underline{\mathbf{X}}}(\mathbf{y}_k | \underline{\mathbf{x}}_k) f_{\underline{\mathbf{X}}|k-1}(\underline{\mathbf{x}}_k | \mathbf{y}_{1:k-1})}{\int f_{\mathbf{Y}|\underline{\mathbf{X}}}(\mathbf{y}_k | \underline{\mathbf{x}}_k) f_{\underline{\mathbf{X}}|k-1}(\underline{\mathbf{x}}_k | \mathbf{y}_{1:k-1}) \delta \underline{\mathbf{x}}_k} \end{aligned} \quad (23.75)$$

where $f_{k|k-1}(\mathbf{x}_k | \underline{\mathbf{x}}_{k-1})$ is the transition density, $f_{\mathbf{Y}|\underline{\mathbf{X}}}(\mathbf{y}_k | \underline{\mathbf{x}}_k)$ is the multi-target likelihood function, and the set integrals are defined as previously, but now also over all labels.

Furthermore, as part of this recursion, one requires the definition of an **association map** at time k as the function $\vartheta : \mathbb{L}_{0:k} \rightarrow \{0, 1, \dots, M(k)\}$ such that $\vartheta(\ell) = \vartheta(\ell') > 0$ implies $\ell = \ell'$. Such a function can be regarded as an assignment of labels to measurements, with undetected labels assigned to 0. The set of all such association maps is denoted as Θ_k . Thus, the subset of association maps with domain L is denoted by $\Theta_k(L)$.

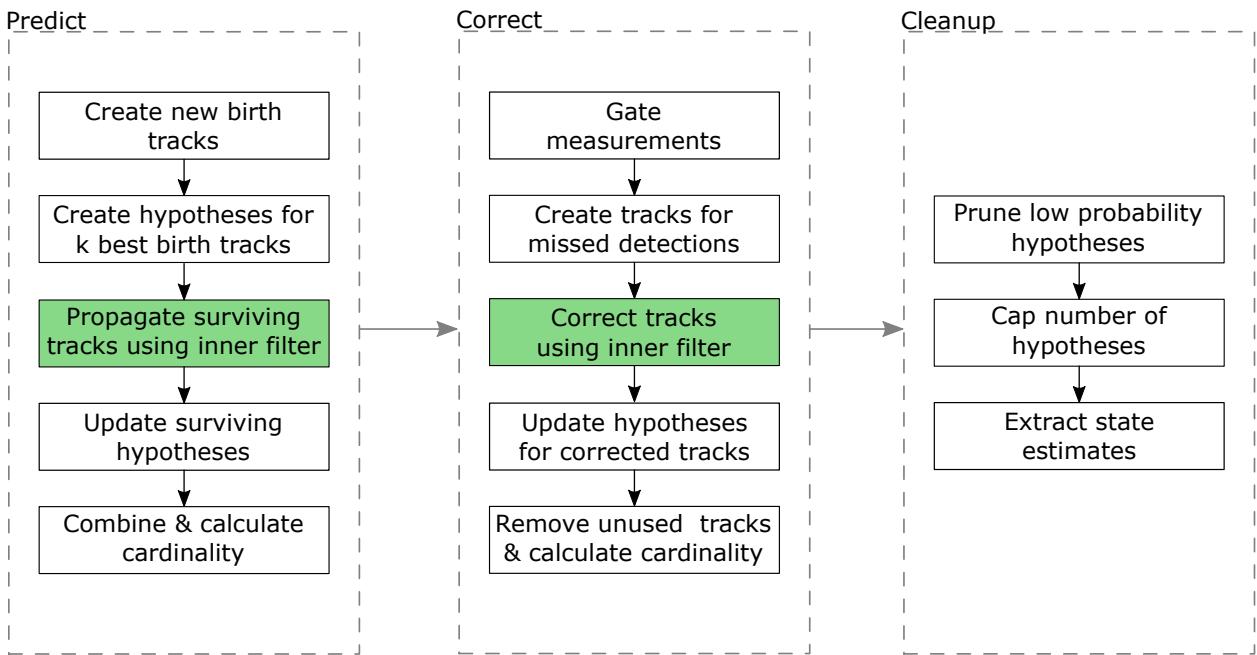
Then, with the assumption that the initial multi-target density is in the GLMB RFS, then the *a priori* and *a posteriori* densities are also in the GLMB RFS. Thus, a closed-form solution to the multi-target Bayes filter can be realized with a GLMB formulation. However, for numerical implementation, an alternative form of the GLMB known as the **δ -generalized labeled multi-Bernoulli (GLMB) RFS**, i.e.

$$f_{\underline{\mathbf{X}}}(\underline{\mathbf{x}}) = \Delta(\underline{\mathbf{x}}) \sum_{(I, \xi) \in \mathcal{F}(\mathbb{L}) \times \Xi} w^{(I, \xi)} \delta_I(\mathcal{L}(\underline{\mathbf{x}})) [f_{\underline{\mathbf{X}}}^{(\xi)}]^\underline{\mathbf{x}} \quad (23.76)$$

which allows for numerical implementation of the labeling scheme presented above, by noting Ξ is the space of association map histories, $\Theta_{0:k} = \Theta_0 \times \dots \times \Theta_k$.

Here, each $I \in \mathcal{F}(\mathbb{L}_{0:k})$ represents a set of tracks at time step k , and each $\xi = (\vartheta_0, \dots, \vartheta_k) \in \Theta_{0:k}$ represents a history of association maps up to time step k , which also contains the history of target labels incorporating target births and deaths. The pair $(I, \xi) \in \mathcal{F}(\mathbb{L}_{0:k}) \times \Theta_{0:k}$ is a **hypothesis** and its associated weight $w^{(I, \xi)}$ is the hypothesis probability. The density $f_{\vec{X}}^{(\xi)}(\cdot, \ell)$ represents the kinematic state of track ℓ for association map history ξ . Thus, one typical state extraction step for GLMB-based filters is to determine the MAP cardinality estimate \hat{N} , determine the MAP label set among those with cardinality \hat{N} , then extract the expected values of the states from the kinematic densities.

The GLMB filter can be outlined as follows using the bookkeeping outer filter for the hypotheses probabilities and cardinality and the inner filter for the kinematic filtering of each hypothesis.



In this fashion, the **δ -GLMB** filter can be implemented with cubic complexity in the number of measurements. This multi-target tracker also enjoys a number of nice analytical properties, e.g. it is a necessary and sufficient statistic of a GLMB and minimizes the L_1 -distance between a GLMB through its truncation, which both can be computed in closed-form. In addition, the simpler **labeled multi-Bernoulli (LMB) filter** was developed which propagated only one term in the label set significantly reducing the runtime of the GLMB filter equations. Lastly, the **joint generalized labeled multi-Bernoulli (JGLMB) filter** was developed in 2016 which combines the prediction and correction step steps of the GLMB into a single recursion with only one truncation step necessary. This JGLMB filter has sped up the LRFS-based filter considerably and is considered the state-of-the-art in multi-target tracking.

23.6 Extended Object Tracking

23.7 Simultaneous Localization and Mapping

Part V

Appendices

A

Fundamental Mathematical Concepts

A.1 Concepts from Set Theory

A mathematical **set** is a collection of points denoted by a boldface capital letter or curly brackets, e.g.

$$\mathbf{S} = \{\dots, \omega, \dots\} \quad (\text{A.1})$$

where ω is an **element**, also known as a **member**, of \mathbf{S} and is denoted by $\omega \in \mathbf{S}$. Of special note is the blackboard bold notation for the sets of the **natural numbers** $0, 1, 2, 3, \dots$, also known as the **counting numbers**, denoted by \mathbb{N} , the **integers** $\dots, -3, -2, -1, 0, 1, 2, 3, \dots$ denoted by \mathbb{Z} , the **rational numbers** denoted by \mathbb{Q} , the **real numbers** denoted by \mathbb{R} , the real numbers with ∞ denoted by \mathbb{R}_+ , and the **complex numbers** denoted by \mathbb{C} .

The **cardinality** of set \mathbf{S} , denoted by $|\mathbf{S}| \geq 0$, is the number of elements in a set \mathbf{S} . Cardinality may be finite or infinite. The **empty set**, denoted by \emptyset , has cardinality zero and is also known as the **null set**. A set \mathbf{S} is a **finite set** if its cardinality is finite. A nonempty set is a **countably infinite set** if the elements $\omega \in \mathbf{S}$ have a one-to-one correspondence with the natural numbers, \mathbb{N} , i.e., one can construct some infinite sequence, $\omega_0, \omega_1, \omega_2, \dots$ where ω_i are distinct and indexed by \mathbb{N} . A nonempty set \mathbf{S} is **countable** if \mathbf{S} is a finite set or a countably infinite set.

Let \mathbf{A} be another set of elements. If every point in \mathbf{A} also belongs contained within \mathbf{S} , then \mathbf{A} is a **subset** of \mathbf{S} denoted by $\mathbf{A} \subseteq \mathbf{S}$. If $\mathbf{S} \subseteq \mathbf{A}$ also, then $\mathbf{S} = \mathbf{A}$. If $\mathbf{S} \not\subseteq \mathbf{A}$, then $\mathbf{A} \subset \mathbf{S}$ is a **proper subset** as $\mathbf{A} = \mathbf{S}$ is not possible.

For $\mathbf{A} \subset \mathbf{S}$ and $\omega \in \mathbf{S}, \omega \notin \mathbf{A}$ denotes that ω is not in \mathbf{A} . The **complement** of \mathbf{A} , denoted \mathbf{A}^c , is defined as

$$\mathbf{A}^c = \{\omega \in \mathbf{S} : \omega \notin \mathbf{A}\} \quad (\text{A.2})$$

Note that $\mathbf{S}^c = \emptyset$ and for any $\mathbf{A} \subset \mathbf{S}, \emptyset \in \mathbf{A}$.

A **function**, denoted by $f : \mathbf{X} \rightarrow \mathbf{Y}$, consists of a set \mathbf{X} of admissible inputs called the **domain**, a set \mathbf{Y} called the **co-domain**, and a **mapping** f that associates a value $f(x) \in \mathbf{Y}$ to each $x \in \mathbf{X}$. This is typically

stated as “ f maps \mathbf{X} into \mathbf{Y} .” The **range** is the set of all possible values of $f(x)$, i.e.,

$$\{f(x) : x \in \mathbf{X}\} \subseteq \mathbf{Y} \quad (\text{A.3})$$

where the range may not be equal to the co-domain. A function is **onto** if for every $y \in \mathbf{Y}$, the equation $f(x) = y$ has *at least* one solution, i.e., $\{f(x) : x \in X\} = \mathbf{Y}$. A function is **one-to-one** if for every $y \in \mathbf{Y}$, the equation $f(x) = y$ has *at most* one solution. A function is **invertible** if for every $y \in \mathbf{Y}$, the equation $f(x) = y$ has one *unique* solution, i.e., it is onto and one-to-one. Two functions are the same if and only if they have the same domain, co-domain, and mapping. Two mappings from the same domain and co-domain are the same if and only if they associate the same value to each $x \in \mathbf{X}$.

The **intersection** of two subsets $\mathbf{A} \subseteq \mathbf{S}$ and $\mathbf{B} \subseteq \mathbf{S}$ is

$$\mathbf{A} \cap \mathbf{B} = \{\omega \in \mathbf{S} : \omega \in \mathbf{A} \text{ and } \omega \in \mathbf{B}\} \quad (\text{A.4})$$

which can be generalized to collections of subsets, $\mathbf{A}_1, \mathbf{A}_2, \dots \subseteq \mathbf{S}$, as

$$\bigcap_i \mathbf{A}_i = \{\omega \in \mathbf{S} : \omega \in \mathbf{A}_i \forall i\} \quad (\text{A.5})$$

Note that if $\mathbf{A} \subseteq \mathbf{B}$, then $\mathbf{A} \cap \mathbf{B} = \mathbf{A}$. Two subsets $\mathbf{A} \subseteq \mathbf{S}$ and $\mathbf{B} \subseteq \mathbf{S}$ are **disjoint** or **mutually exclusive** if

$$\mathbf{A} \cap \mathbf{B} = \emptyset \quad (\text{A.6})$$

and the infinite collection of subsets $\mathbf{A}_1, \mathbf{A}_2, \dots \subseteq \mathbf{S}$ are **pairwise disjoint** or **mutually exclusive** if

$$\mathbf{A}_i \cap \mathbf{A}_j = \emptyset \quad \forall i \neq j \quad (\text{A.7})$$

The **union** of two subsets $\mathbf{A} \subseteq \mathbf{S}$ and $\mathbf{B} \subseteq \mathbf{S}$ is

$$\mathbf{A} \cup \mathbf{B} = \{\omega \in \mathbf{S} : \omega \in \mathbf{A} \text{ or } \omega \in \mathbf{B}\} \quad (\text{A.8})$$

with “or” inclusive which can be generalized to collections of subsets, $\mathbf{A}_1, \mathbf{A}_2, \dots \subseteq \mathbf{S}$, as

$$\bigcup_i \mathbf{A}_i = \{\omega \in \mathbf{S} : \omega \in \mathbf{A}_i \text{ for some } i\} \quad (\text{A.9})$$

A **partition** is a collection of pairwise disjoint, nonempty sets, $\mathbf{A}_1, \mathbf{A}_2, \dots \subseteq \mathbf{S}$, with

$$\bigcup_i \mathbf{A}_i = \mathbf{S} \quad (\text{A.10})$$

The **set difference** to two subsets $\mathbf{A} \subseteq \mathbf{S}$ and $\mathbf{B} \subseteq \mathbf{S}$ is

$$\mathbf{B} \setminus \mathbf{A} = \mathbf{B} \cap \mathbf{A}^c \quad (\text{A.11})$$

The **commutative laws** for two subsets $\mathbf{A} \subseteq \mathbf{S}$ and $\mathbf{B} \subseteq \mathbf{S}$ are

$$\mathbf{A} \cap \mathbf{B} = \mathbf{B} \cap \mathbf{A} \quad \text{and} \quad \mathbf{A} \cup \mathbf{B} = \mathbf{B} \cup \mathbf{A} \quad (\text{A.12})$$

The **associative laws** for three subsets $\mathbf{A} \subseteq \mathbf{S}$, $\mathbf{B} \subseteq \mathbf{S}$, and $\mathbf{C} \subseteq \mathbf{S}$ are

$$\mathbf{A} \cap (\mathbf{B} \cap \mathbf{C}) = (\mathbf{A} \cap \mathbf{B}) \cap \mathbf{C} \quad \text{and} \quad \mathbf{A} \cup (\mathbf{B} \cup \mathbf{C}) = \mathbf{A} \cup (\mathbf{B} \cup \mathbf{C}) \quad (\text{A.13})$$

The **distributive laws** for three subsets $\mathbf{A} \subseteq \mathbf{S}$, $\mathbf{B} \subseteq \mathbf{S}$, and $\mathbf{C} \subseteq \mathbf{S}$ are

$$\mathbf{A} \cap (\mathbf{B} \cup \mathbf{C}) = (\mathbf{A} \cap \mathbf{B}) \cup (\mathbf{A} \cap \mathbf{C}) \quad \text{and} \quad \mathbf{A} \cup (\mathbf{B} \cap \mathbf{C}) = (\mathbf{A} \cup \mathbf{B}) \cap (\mathbf{A} \cup \mathbf{C}) \quad (\text{A.14})$$

which can be generalized to infinite collections of subsets, $\mathbf{A}, \mathbf{B}_1, \mathbf{B}_2, \dots \subseteq \mathbf{S}$, as

$$\mathbf{A} \cap \left(\bigcup_{i=1}^{\infty} \mathbf{B}_i \right) = \bigcup_{i=1}^{\infty} (\mathbf{A} \cap \mathbf{B}_i) \quad \text{and} \quad \mathbf{A} \cup \left(\bigcap_{i=1}^{\infty} \mathbf{B}_i \right) = \bigcap_{i=1}^{\infty} (\mathbf{A} \cup \mathbf{B}_i) \quad (\text{A.15})$$

De Morgan's laws for three subsets $\mathbf{A} \subseteq \mathbf{S}$, $\mathbf{B} \subseteq \mathbf{S}$, and $\mathbf{C} \subseteq \mathbf{S}$ are

$$(\mathbf{A} \cap \mathbf{B})^c = \mathbf{A}^c \cup \mathbf{B}^c \quad \text{and} \quad (\mathbf{A} \cup \mathbf{B})^c = \mathbf{A}^c \cap \mathbf{B}^c \quad (\text{A.16})$$

which can be generalized to infinite collections of subsets, $\mathbf{A}_1, \mathbf{A}_2, \dots \subseteq \mathbf{S}$, as

$$\left(\bigcap_{i=1}^{\infty} \mathbf{A}_i \right)^c = \bigcup_{i=1}^{\infty} \mathbf{A}_i^c \quad \text{and} \quad \left(\bigcup_{i=1}^{\infty} \mathbf{A}_i \right)^c = \bigcap_{i=1}^{\infty} \mathbf{A}_i^c \quad (\text{A.17})$$

References

For more information, please refer to the following

- Gubner, J. A., “1.2 Review of set notation,” in *Probability and Random Processes for Electrical and Computer Engineers*, Cambridge University Press, 2006

A.2 Concepts from Linear Algebra

Vector and Matrix Definitions

This textbook denotes **scalars** by unbolded, block lowercase letters, e.g. x .

This textbook denotes **vectors** with **dimension** n_x by unbolded block lowercase letters with an arrow above, e.g., \vec{x} , which takes on values from some set \mathbf{Y} , denoted as $\vec{x} \in \mathbf{Y}^{n_x}$, and is represented in a bracketed column as

$$\vec{x} = \begin{bmatrix} x_1 \\ \vdots \\ x_{n_x} \end{bmatrix} \quad (\text{A.18})$$

where x_i denotes the scalar **element** of \vec{x} in the i^{th} row. \mathbf{Y} is a **vector space** defined as a set of vectors whose members can be added and scaled and still belong to the vector space. A **real vector** with dimension n_x takes on n_x real number values and is denoted as $\vec{x} \in \mathbb{R}^{n_x}$. A **complex vector** with dimension n_x takes on

n_x complex number values and is denoted as $\vec{x} \in \mathbb{C}^{n_x}$. Note that this notation exclusively uses vectors as **columns** where a **row vector** is denoted by the **transpose** $\vec{x}^T \in \mathbb{R}^{n_x}$, i.e.,

$$\vec{x}^T = [x_1 \quad \cdots \quad x_{n_x}] \quad (\text{A.19})$$

This textbook denotes **matrices** by unbolded, block capital letters, e.g., A , which takes on values from some set \mathbf{Y} with a **row dimension** m and a **column dimension**, n , is denoted as $A \in \mathbf{Y}^{m \times n}$, and is represented in a bracketed table as

$$A = \begin{bmatrix} A_{1,1} & \cdots & A_{1,n} \\ \vdots & \ddots & \vdots \\ A_{m,1} & \cdots & A_{m,n} \end{bmatrix} \quad (\text{A.20})$$

where $A_{i,j}$ denotes the scalar **element** of A in the i^{th} row and j^{th} column. A **real matrix** with m rows and n columns takes on mn real number values and is denoted as $A \in \mathbb{R}^{m \times n}$. A **complex matrix** with m rows and n columns takes on mn complex values and is denoted as $A \in \mathbb{C}^{m \times n}$.

A is a **square matrix** if it has an equal number of rows and columns.

The **matrix transpose** of A is $A^T = B$ which assigns $B_{i,j} = A_{j,i}$.

The **conjugate matrix transpose** or **Hermitian transpose** of A is $A^* = A^H = B$ which assigns $B_{i,j} = A_{j,i}^*$.

The **matrix inverse** of A is the solution to $A^{-1}A = I$.

A useful **matrix inversion lemma** for the inverse of the sum of two matrices is

$$(A + BCD)^{-1} = A^{-1} - A^{-1}B(C^{-1} + DA^{-1}C)^{-1}DA^{-1} = A^{-1}B(C^{-1} + DA^{-1}C)^{-1}(BC)^{-1} \quad (\text{A.21})$$

A is an **orthogonal matrix** if $A^{-1} = A^T$.

A is an **unitary matrix** if $A^{-1} = A^*$.

A is a **symmetric matrix** if $A = A^T$.

A is a **Hermitian matrix** if $A = A^*$.

A is a **diagonal matrix** if for $i \neq j$, $A_{i,j} = 0$.

A is an **upper triangular matrix** if for $i > j$, $A_{i,j} = 0$.

A is a **lower triangular matrix** if for $i < j$, $A_{i,j} = 0$.

The **rank** of a matrix M , denoted by $\text{rank}(M)$, is a measure of the “non-degenerateness” of the system of linear equations encoded by M . More formally, this can be expressed mathematically as the maximal number of linearly independent rows/columns of M or as the **dimension of the vector space** spanned by the rows/columns of M . Vector spaces are the formal subject of linear algebra and are well characterized by its dimension, or the number of linearly independent “directions” in the space. Thus, the rank of a matrix can describe the dimension of the column space, i.e. linearly independent columns, of M as the **column rank** and the dimension of the row space, i.e. linearly independent rows, of M as the **row rank**. A fundamental theorem of linear algebra is that column rank is equal to row rank. Lastly, M is said to have **full rank** if its rank equals the lesser of the number of rows and columns, i.e. $\text{rank}(M) = \min(m, n)$. Conversely, M is said to be **rank deficient** if it does not have full rank.

Vector and Matrix Norms

The **vector norm**, $\|\vec{x}\|$, of any vector $\vec{x} \in \mathbb{R}^{n_x}$ is a real valued function from $\mathbb{R}^{n_x} \rightarrow \mathbb{R}$ with the following properties

1. $\|\vec{x}\| \geq 0$;
2. $\|\vec{x}\| = 0$ if and only if \vec{x} is the zero vector in \mathbb{R}^{n_x} ;
3. for any $\lambda \in \mathbb{R}$, $\|\lambda \vec{x}\| = |\lambda| \|\vec{x}\|$; and
4. for any $\vec{y} \in \mathbb{R}^{n_x}$, the **triangle inequality**

$$\|\vec{x} + \vec{y}\| \leq \|\vec{x}\| + \|\vec{y}\| \quad (\text{A.22})$$

holds.

An important class of vector norms are the **L_p -norms**, also known as **p -norms**, defined as

$$\|\vec{x}\|_p = \left(\sum_{i=1}^{n_x} |x_i|^p \right)^{\frac{1}{p}}, \quad 1 \leq p \leq \infty \quad (\text{A.23})$$

Of particular interest are the **L_1 -norm**, also known as the **taxicab norm**,

$$\|\vec{x}\|_1 = \sum_{i=1}^{n_x} |x_i| \quad (\text{A.24})$$

the **L_2 -norm**, also known as the **Euclidean norm**,

$$\|\vec{x}\|_2 = \left(\sum_{i=1}^{n_x} x_i^2 \right)^{\frac{1}{2}} = \sqrt{\vec{x}^T \vec{x}} \quad (\text{A.25})$$

and the **L_∞ -norm**, also known as the **vector maximum norm**,

$$\|\vec{x}\|_\infty = \max |x_i| \quad (\text{A.26})$$

As an aside, another popular norm is the **weighted L_2 -norm**,

$$\|\vec{x}\|_W = \sqrt{\vec{x}^T W \vec{x}} \quad (\text{A.27})$$

where W is known as the **weight matrix**.

The **matrix norm**, $\|A\|$, of the matrix $A \in \mathbb{R}^{m \times n}$ is a real valued function from $\mathbb{R}^{m \times n} \rightarrow \mathbb{R}$ with the following properties

1. $\|A\| \geq 0$;
2. $\|A\| = 0$ if and only if A is the zero matrix in $\mathbb{R}^{m \times n}$;
3. for any $\lambda \in \mathbb{R}$, $\|\lambda A\| = |\lambda| \|A\|$; and
4. for any $B \in \mathbb{R}^{m \times n}$, the **triangle inequality**

$$\|A + B\| \leq \|A\| + \|B\| \quad (\text{A.28})$$

holds.

An important class of matrix norms are **induced matrix norms** which are defined for a matrix A and some specified norm $\|\vec{x}\|$ as

$$\|A\| = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|}{\|\vec{x}\|} \quad (\text{A.29})$$

where sup stands for the **supremum**, also known as the **least upper bound** of the specified set. Typically one uses L^p -norms for induced matrix norms which can be written as

$$\|A\|_p = \sup_{\vec{x} \neq 0} \frac{\|A\vec{x}\|_p}{\|\vec{x}\|_p} \quad (\text{A.30})$$

Another important class of matrix norms are **entry-wise matrix norms** which treats a matrix A as a vector of size $m \times n$ and uses a vector norm. One such norm is the $L_{p,q}$ -norms defined as

$$\|A\|_{p,q} = \left(\sum_{j=1}^n \left(\sum_{i=1}^m |A_{i,j}|^p \right)^{\frac{q}{p}} \right)^{\frac{1}{q}} \quad (\text{A.31})$$

where $A_{i,j}$ denotes the element of A in the i^{th} row and j^{th} column. The $L_{2,2}$ -norm is also known as the **Frobenius norm** and can be defined as

$$\|A\|_{2,2} = \|A\|_F = \left(\sum_{i=1}^m \sum_{j=1}^n |A_{i,j}|^2 \right)^{\frac{1}{2}} \quad (\text{A.32})$$

In addition the $L_{\infty,\infty}$ -norm is also known as the **matrix maximum norm** and can be defined as

$$\|A\|_{\max} = \max_{i,j} |A_{i,j}| \quad (\text{A.33})$$

Matrix Decompositions

Eigenvalue decomposition of the state matrix. A **eigenvalue**, λ , with associated **(left) eigenvector**, \vec{v} , of a square matrix A is defined as a solution to the **eigenvalue problem** defined as “given some A , for what values of λ and \vec{v} does the following equation hold:”

$$A\vec{v} = \lambda\vec{v} \quad (\text{A.34})$$

To solve this problem, one can rewrite the equation above as

$$\lambda\vec{v} - A\vec{v} = 0 \quad (\text{A.35})$$

$$[\lambda I - A]\vec{v} = 0 \quad (\text{A.36})$$

which for nontrivial solutions, i.e. $\vec{v} \neq 0$, one must solve for

$$\det(\lambda I - A) = 0 \quad (\text{A.37})$$

to obtain the eigenvalues of A , i.e. $\lambda(A)$. Then, by substituting back into $[\lambda I - A]\vec{v} = 0$, one can also obtain the associated (left) eigenvectors.

In general, λ and \vec{v} can be complex-valued even if A is real-valued. Based on the values of λ , one can define the following:

- A is **positive definite**, denoted by $A > 0$, if the real parts of all $\lambda(A)$ are > 0 ;
- A is **positive semi-definite**, denoted by $A \geq 0$, if the real parts of all $\lambda(A)$ are ≥ 0 ;
- A is **negative definite**, denoted by $A < 0$, if the real parts of all $\lambda(A)$ are < 0 ;
- A is **negative semi-definite**, denoted by $A \leq 0$, if the real parts of all $\lambda(A)$ are ≤ 0 ; and
- A is **indefinite** otherwise.

It should also be noted that a negative definite A matrix is also known as a **stable matrix** or **Hurwitz matrix**.

Furthermore, a square matrix A with dimensions $n \times n$ is **diagonalizable**, if an eigenvalue decomposition can be performed as

$$A = V\Lambda V^{-1} \quad (\text{A.38})$$

where the n corresponding *non-repeated* eigenvalues of A makeup a diagonal Λ as

$$\Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} \quad (\text{A.39})$$

the n linearly independent right eigenvectors of A makeup V as

$$V = [\vec{v}_1 \quad \cdots \quad \vec{v}_n] \quad (\text{A.40})$$

and the n linearly independent left eigenvectors of A makeup V^{-1} as

$$V^{-1} = \begin{bmatrix} \vec{\mu}_1^T \\ \vdots \\ \vec{\mu}_n^T \end{bmatrix} \quad (\text{A.41})$$

However, if there are *repeated* eigenvalues of A , then A may not be diagonalizable. Then, a **Jordan matrix** can be used to form Λ in a similar fashion to an eigenvalue decomposition where a Jordan matrix, J , is defined as

$$J = \begin{bmatrix} J_1 & 0 & \cdots & 0 \\ 0 & J_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & J_k \end{bmatrix} \quad (\text{A.42})$$

where the 0's are zero-valued matrices and the k **Jordan blocks**, J_k , are specified by dimension r and eigenvalue λ_r , i.e.

$$J_k(r, \lambda_r) = \begin{bmatrix} \lambda_r & 1 & 0 & \cdots & 0 & 0 \\ 0 & \lambda_r & 1 & \cdots & 0 & 0 \\ 0 & 0 & \lambda_r & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & \lambda_r & 1 \\ 0 & 0 & 0 & \cdots & 0 & \lambda_r \end{bmatrix} \quad (\text{A.43})$$

For diagonalizable matrices, the Jordan matrix is purely diagonal since each constituent Jordan block is 1×1 . The Jordan matrix is useful in forming the **Jordan Canonical Form (JCF)** of LTI state-space systems through the substitution

$$\vec{x} = V \vec{z} \quad (\text{A.44})$$

which allows the continuous-time LTI state-space system to be rewritten as

$$\begin{aligned} V \dot{\vec{z}}(t) &= AV \vec{z}(t) + B \vec{u}(t) \\ \vec{y}(t) &= CV \vec{z}(t) + D \vec{u}(t) \end{aligned} \quad (\text{A.45})$$

$$\begin{aligned} \dot{\vec{z}}(t) &= V^{-1} AV \vec{z}(t) + V^{-1} B \vec{u}(t) \\ \vec{y}(t) &= CV \vec{z}(t) + D \vec{u}(t) \end{aligned} \quad (\text{A.46})$$

$$\begin{aligned} \dot{\vec{z}}(t) &= \Lambda \vec{z}(t) + \bar{B} \vec{u}(t) \\ \vec{y}(t) &= \bar{C} \vec{z}(t) + D \vec{u}(t) \end{aligned} \quad (\text{A.47})$$

where Λ is a Jordan matrix (i.e. diagonal or nearly diagonal), and \bar{B} and \bar{C} are new input and output matrices, respectively. To obtain V , one must solve for the **generalized eigenvectors** for each Jordan block, $J_k(r, \lambda_r)$, which solve

$$\begin{aligned} (A - \lambda_r I) \vec{v}_1 &= 0 \\ (A - \lambda_r I) \vec{v}_2 &= \vec{v}_1 \\ &\vdots \\ (A - \lambda_r I) \vec{v}_r &= \vec{v}_{r-1} \end{aligned} \quad (\text{A.48})$$

When considering the full LTI system behavior from input to output, one often uses the **singular value decomposition (SVD)** defined for any $m \times n$ real-valued matrix M as

$$M = U \Sigma V^{-1} \quad (\text{A.49})$$

where U is an $m \times m$ orthogonal matrix, Σ is a diagonal $m \times n$ matrix with *non-negative* real numbers on the diagonal, i.e.

$$\Sigma = \begin{bmatrix} \Sigma_1 & \tilde{0} \\ \tilde{0} & \tilde{0} \end{bmatrix}, \quad \Sigma_1 = \text{diag}(\sigma_1, \dots, \sigma_k) \quad (\text{A.50})$$

and V is an $n \times n$ orthogonal matrix. The diagonal entries of Σ are known as the **singular values** of M and are ordered in size with $\sigma = \sigma_1 \geq \dots \geq \sigma_k = \underline{\sigma}$ with $k = \min(m, n)$ singular values because of its non-square nature. In addition, one can say that U and V are orthogonal, $U^{-1} = U^T$ and $V^{-1} = V^T$. Thus the SVD could also be written as

$$M = U \Sigma V^T \quad (\text{A.51})$$

which is much simpler to compute.

The singular values σ_i for M are the non-negative real numbers for which there exists unit-length real-valued vectors \vec{u} and \vec{v} such that

$$M \vec{v}_i = \sigma_i \vec{u} \quad (\text{A.52})$$

and

$$\vec{u}_i^T M = \sigma_i \vec{v}^T \quad \text{or} \quad M^T \vec{u}_i = \sigma_i \vec{v} \quad (\text{A.53})$$

where \vec{v}_i is the right-singular vector for σ_i and all together make up the columns of V , i.e.

$$V = [\vec{v}_1 \quad \cdots \quad \vec{v}_n] \quad (\text{A.54})$$

and \vec{u}_i is the left-singular vector for σ_i and all together make up the columns of U , i.e.

$$U = [\vec{u}_1 \quad \cdots \quad \vec{u}_m] \quad (\text{A.55})$$

The singular value problem is similar to the eigenvalue problem except one has two related problems due to the non-square nature of M . The k nonzero singular values of M , $\sigma_i(M)$, can be shown to be related to the eigenvalue decomposition by

$$\sigma_i(M) = \sqrt{\lambda_i(M^* M)} = \sqrt{\lambda_i(M M^*)} > 0 \quad (\text{A.56})$$

and

$$\begin{aligned} M^* M \vec{v}_i &= \sigma_i^2 \vec{v}_i \\ M M^* \vec{u}_i &= \sigma_i^2 \vec{u}_i \end{aligned} \quad (\text{A.57})$$

Thus, all σ_i^2 are eigenvalues of $M M^*$ and $M^* M$, all \vec{v}_i are eigenvectors of $M^* M$, and all \vec{u}_i are eigenvectors of $M M^*$.

The **maximum singular value** of A can be shown to be

$$\bar{\sigma}(A) = \max_{\vec{x} \neq 0} \frac{\|A \vec{x}\|_2}{\|\vec{x}\|_2} = \|A\|_2 \quad (\text{A.58})$$

while the **minimum singular value** of A can be shown to be

$$\underline{\sigma}(A) = \min_{\vec{x} \neq 0} \frac{\|A \vec{x}\|_2}{\|\vec{x}\|_2} \quad (\text{A.59})$$

The $\bar{\sigma}(A)$, i.e. $\|A\|_2$, represents how “big” A is or how large the “gain” of A is. The $\underline{\sigma}(A)$ represents how nearly singular A is. The **condition number** for A is defined as

$$\kappa(A) = \frac{\bar{\sigma}(A)}{\underline{\sigma}(A)} \quad (\text{A.60})$$

and can be used to determine how invertible A is.

Lastly, there are other matrix decompositions that may be used in computational algorithms for linear state-space systems. One such decomposition is the **QR decomposition** for any square matrix A defined as

$$A = QR \quad (\text{A.61})$$

where Q is orthogonal and R is upper triangular. A second decomposition is the **Cholesky decomposition** for Hermitian positive definite matrices defined as

$$A = LL^* \quad (\text{A.62})$$

where L is an upper triangular matrix with real and positive entries along its main diagonal. Note that if A is real-valued, then this reduces to $A = LL^T$.

Linear Matrix Equalities and Inequalities

A **linear matrix equality (LME)** in the variable $X \in \mathcal{X}$ is an equality of the form

$$F(X) = Q \quad (\text{A.63})$$

and a **linear matrix inequality (LMI)** in the variable $X \in \mathcal{X}$ is an inequality of the form

$$F(X) \leq Q \quad (\text{A.64})$$

where \mathcal{X} is a real-valued vector space, F is a *linear* mapping from $\mathcal{X} \rightarrow \mathbb{H}^n$, and $Q \in \mathbb{H}^n$ where \mathbb{H}^n is the space of $n \times n$ Hermitian matrices. Furthermore, a **strict linear matrix inequality (LMI)** in $X \in \mathcal{X}$ is an inequality of the form

$$F(X) < Q \quad (\text{A.65})$$

An alternative form for LMEs and LMIs can be formulated using vectors by defining $\vec{v}_1, \dots, \vec{v}_m$ as the basis for the vector space, \mathcal{X} . Next, for any $X \in \mathcal{X}$, one has that there exists scalars x_1, \dots, x_m such that

$$X = x_1 \vec{v}_1 + \dots + x_m \vec{v}_m \quad (\text{A.66})$$

Then, by the linearity of $F(X)$, one has for LMEs

$$x_1 F(\vec{v}_1) + \dots + x_m F(\vec{v}_m) = Q \quad (\text{A.67})$$

and for LMIs

$$x_1 F(\vec{v}_1) + \dots + x_m F(\vec{v}_m) \leq Q \quad (\text{A.68})$$

where the variables become the set of scalars: x_1, \dots, x_m . However, this coordinate form is not typically how one encounters LMIs in control theory. Thus, to check whether an inequality is an LMI, one can simply check whether the three conditions above are satisfied.

While many matrix inequalities may not look like LMIs at first glance, consider the following matrix inequality

$$A^* X A - X = Q \quad (\text{A.69})$$

and

$$A^* X A - X < Q \quad (\text{A.70})$$

with $A \in \mathbb{R}^{n \times n}$, $X \in \mathbb{S}^n$, and $Q \in \mathbb{H}^n$ where \mathbb{S}^n is the space of $n \times n$ symmetric matrices. Defining $F(X) = A^* X A - X$, one can clearly see this is a linear mapping in the space of symmetric matrices as $F(X)\mathbb{S}^n$. Next, consider

$$A^* X A - B Y + Y^* B^* = Q \quad (\text{A.71})$$

and

$$A^* X A - B Y + Y^* B^* < Q \quad (\text{A.72})$$

with $A \in \mathbb{C}^{n \times n}$, $B \in \mathbb{C}^{n \times m}$, $Q \in \mathbb{H}^n$, $X \in \mathbb{S}^n$, and $Y \in \mathbb{R}^{m \times n}$. Defining $Z = (X, Y)$ and $F(Z) = A^* X A - B Y + Y^* B^*$, one can see this is LME/LMI in X and Y as $F(Z) : \mathbb{S}^n \times \mathbb{R}^{m \times n} \rightarrow \mathbb{H}^n$ is a linear mapping. Furthermore, it should be noted that in some problems LMEs/LMIs are written over the space \mathbb{S}^n instead of \mathbb{H}^n .

A useful fact for LMEs/LMIs is that a list of LMEs/LMIs can always be converted to a *single* LMI through block diagonal matrices, e.g. two LMIs:

$$F_1(X_1) < Q_1 \quad (\text{A.73})$$

and

$$F_2(X_2) < Q_2 \quad (\text{A.74})$$

are equivalent to one LMI

$$F(X) = \begin{bmatrix} F_1(X_1) & 0 \\ 0 & F_2(X_2) \end{bmatrix} < \begin{bmatrix} Q_1 & 0 \\ 0 & Q_2 \end{bmatrix} = Q \quad (\text{A.75})$$

Another useful fact for deriving LMIs is to use the **Schur Complement Lemma** which states for matrices: $A \in \mathbb{H}^n$, $B \in \mathbb{H}^n$, and C , the matrix inequality

$$\begin{bmatrix} A & C \\ C^* & B \end{bmatrix} < 0 \quad (\text{A.76})$$

is satisfied if and only if

$$\begin{bmatrix} A - CB^{-1}C^* & 0 \\ 0 & B \end{bmatrix} < 0 \quad (\text{A.77})$$

also is satisfied.

To prove this lemma, one can left and right multiply this second inequality by the matrix

$$M = \begin{bmatrix} I & CB^{-1} \\ 0 & I \end{bmatrix} \quad (\text{A.78})$$

and its Hermitian transpose to obtain

$$\begin{bmatrix} A & C \\ C^* & B \end{bmatrix} = \begin{bmatrix} I & CB^{-1} \\ 0 & I \end{bmatrix} \begin{bmatrix} A - CB^{-1}C^* & 0 \\ 0 & B \end{bmatrix} \begin{bmatrix} I & 0 \\ B^{-1}C^* & I \end{bmatrix} < 0 \quad (\text{A.79})$$

where the right side uses the fact that if a matrix $Q < 0$, then by definition $MQM^* < 0$.

References

For more information, please refer to the following

- Henderson, H. V., “On Deriving the Inverse of a Sum of Matrices,” in *Society for Industrial and Applied Mathematics*, Vol. 23, No. 1, 1981, pp. 53-60

A.3 Concepts from Complex Analysis and Stability Criterion

Nyquist Plot

The **Nyquist plot** is a common tool used to understand the stability and robustness of a feedback control system. Similar to the Bode plot, the Nyquist plot can be used to analyze the frequency response of a transfer function, i.e. $G(s)$ with $s = j\omega$. However, opposed to the Bode plot, the Nyquist plot visualizes the real and imaginary parts of $G(j\omega)$ as a single curve with the real part on the horizontal axis and the imaginary on the vertical axis. It should also be noted the convention for the Nyquist plot is to plot over $-\infty < \omega < \infty$ as opposed to $\omega \geq 0$ for the Bode plot. This convention results in a reflected curve about the real axis for $\omega < 0$ with respect to $\omega > 0$ as a transfer function value $G(j\omega) = \alpha + j\beta$ simply provides the complex conjugate for $\omega < 0$ with respect to $\omega > 0$.

As an example of a Nyquist plot, consider the standard form of a first order LTI system, i.e.

$$\dot{y} + a_0 y = b_0 u \quad (\text{A.80})$$

and assume that $a_0 \neq 0$. The transfer function for this system is $G(s)$

$$G(s) = \frac{b_0}{s + a_0} \quad (\text{A.81})$$

and has a pole at $s = -a_0$.

The Nyquist plot of the frequency response requires computing $G(j\omega)$, i.e.

$$G(j\omega) = \frac{b_0}{j\omega + a_0} \quad (\text{A.82})$$

$$G(j\omega) = \frac{b_0}{j\omega + a_0} \frac{a_0 - j\omega}{a_0 - j\omega} \quad (\text{A.83})$$

$$G(j\omega) = \frac{a_0 b_0}{a_0^2 + \omega^2} - j \frac{\omega b_0}{a_0^2 + \omega^2} \quad (\text{A.84})$$

where

$$\operatorname{Re}\{G(j\omega)\} = \frac{a_0 b_0}{a_0^2 + \omega^2} \quad (\text{A.85})$$

and

$$\operatorname{Im}\{G(j\omega)\} = \frac{-\omega b_0}{a_0^2 + \omega^2} \quad (\text{A.86})$$

Next, noting that

$$\operatorname{Re}\{G(j\omega)\}^2 = \frac{a_0^2 b_0^2}{(a_0^2 + \omega^2)^2} \quad (\text{A.87})$$

$$\operatorname{Im}\{G(j\omega)\}^2 = \frac{\omega^2 b_0^2}{(a_0^2 + \omega^2)^2} \quad (\text{A.88})$$

and adding, one has

$$\operatorname{Re}\{G(j\omega)\}^2 + \operatorname{Im}\{G(j\omega)\}^2 = \frac{a_0^2 b_0^2}{(a_0^2 + \omega^2)^2} + \frac{\omega^2 b_0^2}{(a_0^2 + \omega^2)^2} \quad (\text{A.89})$$

or

$$\operatorname{Re}\{G(j\omega)\}^2 + \operatorname{Im}\{G(j\omega)\}^2 = \frac{b_0^2}{(a_0^2 + \omega^2)} \quad (\text{A.90})$$

Then rearranging, one has

$$\operatorname{Re}\{G(j\omega)\}^2 - \frac{b_0^2}{(a_0^2 + \omega^2)} + \operatorname{Im}\{G(j\omega)\}^2 = 0 \quad (\text{A.91})$$

Then, adding $\left(\frac{b_0}{2a_0}\right)^2$ to both sides, one has

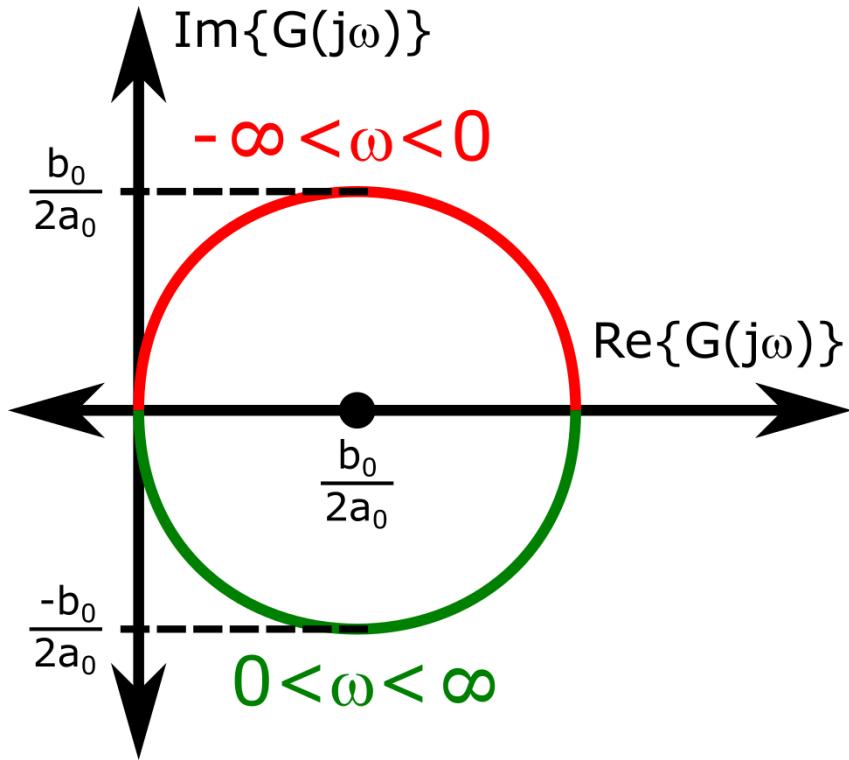
$$\operatorname{Re}\{G(j\omega)\}^2 - \frac{b_0^2}{(a_0^2 + \omega^2)} + \left(\frac{b_0}{2a_0}\right)^2 + \operatorname{Im}\{G(j\omega)\}^2 = \left(\frac{b_0}{2a_0}\right)^2 \quad (\text{A.92})$$

and recalling Equation A.87, one has

$$\left(\operatorname{Re}\{G(j\omega)\} - \frac{b_0}{2a_0}\right)^2 + \operatorname{Im}\{G(j\omega)\}^2 = \left(\frac{b_0}{2a_0}\right)^2 \quad (\text{A.93})$$

which is the equation of a circle of radius $\frac{b_0}{2a_0}$ centered at $\left(\frac{b_0}{2a_0}, 0\right)$.

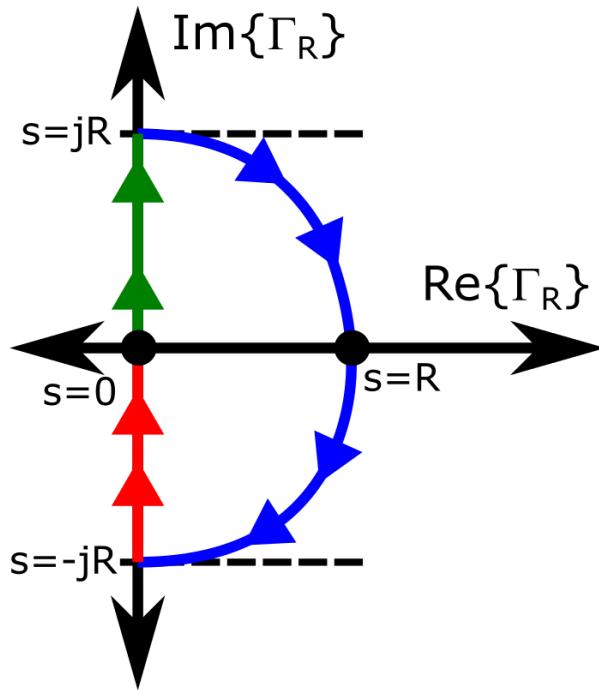
For plotting the variation with ω , consider three particular points. At $\omega = 0$, $G(0) = \frac{b_0}{a_0}$. As $\omega \rightarrow \infty$, $G(j\omega) \rightarrow -j\frac{b_0}{\omega}$. At $\omega = a_0$: $G(ja_0) = (1 - j)\frac{b_0}{2a_0}$. Using these results, one can draw the Nyquist plot as



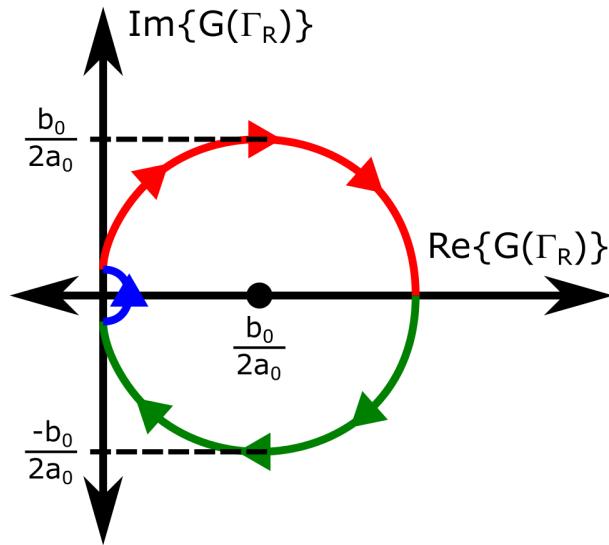
Cauchy's Argument Principle

The Nyquist plot of the open-loop transfer function $L(s)$ can be used to state a theorem concerning the stability of a feedback control system. However, to derive this analysis tool, one requires a result from complex analysis, Cauchy's argument principle, which will be summarized as follows. To begin, let $G(s)$ be a transfer function of a system and let Γ be a simple, closed curve in the complex plane where a “simple” curve is one that does not intersect itself and a closed curve is completely connected. Here, the notation $G(\Gamma)$ is the curve obtained by mapping each complex number $s \in \Gamma$ to another complex number $G(s)$. In general, $G(\Gamma)$ will be closed but need not be simple.

For example, consider the curve Γ_R shown as



which results in the mapping to $G(\Gamma_R)$ for a first order system as



and it can be shown that $G(\Gamma_R)$ converges to the Nyquist plot of $G(s)$ as $R \rightarrow \infty$.

Next, define N_p and N_z as the number of poles and zeros of $G(s)$ inside a general curve Γ , respectively.

Then, **Cauchy's argument principle** states that if Γ does not pass through any poles or zeros of $G(s)$, then the closed curve $G(\Gamma)$ encircles the origin $N_z - N_p$ times. Furthermore, if $N_z - N_p > 0$, the $G(\Gamma)$ encircles the origin clockwise and if $N_z - N_p < 0$, then $G(\Gamma)$ encircles the origin counter-clockwise.

Finally, it should also be noted that if Γ passes through a pole of $G(s)$, then $G(\Gamma)$ diverges to ∞ and if Γ passes through a zero of $G(s)$, then $G(\Gamma)$ intersects the origin. In either case, Cauchy's argument principle would not apply. The proof of this principle can be found elsewhere, e.g. in textbooks on complex analysis, and is left for the reader.

Nyquist Stability Criterion

To develop the stability criteria based on the Nyquist plot of the open-loop transfer function $L(j\omega)$, first note that $s = -1$ is a critical point for the stability of the open-loop transfer function $L(s)$. Recall from previous analysis that the stability of the feedback control system could be assessed through the zeros of $1 + G(s)K(s)$, i.e. $1 + L(s)$ as this expression appeared in the denominator of all four fundamental transfer functions. Thus, if the frequency response of the open-loop transfer function, i.e. $L(j\omega)$, passes through $j\omega = -1$ on the Nyquist plot for some $\omega = \omega_0$, then the feedback control system will be unstable, as any signal that contains frequency ω_0 would cause one or more of the fundamental transfer functions to have infinite gain, e.g. as $S(s) = \frac{E(s)}{Y_C(s)} = \frac{1}{1+L(s)}$, then $|S(j\omega_0)| = \infty$.

Furthermore, recall the stability conditions that if no pole-zero cancellations exist between $G(s)$ and $K(s)$, then the feedback control system is stable if and only if $1 + L(s)$ has no zeros in the RHP. Thus, one can infer that the critical $s = -1$ point also plays a role in the Nyquist theorem on the feedback control system stability.

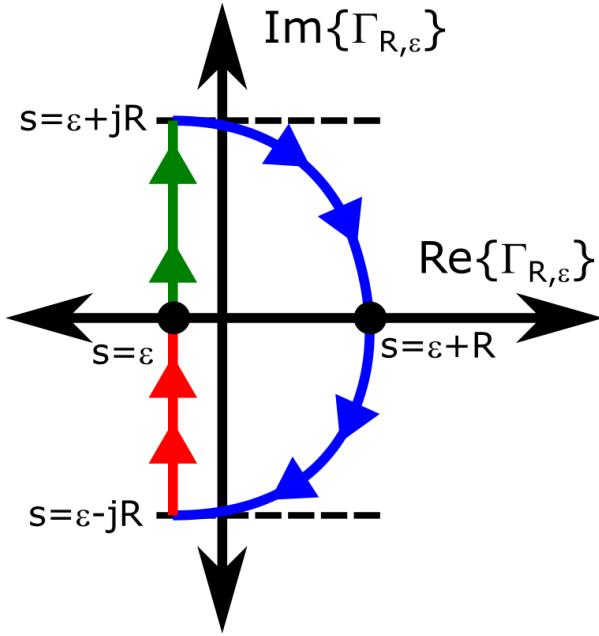
Next, define $P_{c-l}(\epsilon)$ as the number of poles of the closed-loop feedback control system with real part greater than or equal to ϵ , $P_{o-l}(\epsilon)$ as the number of poles of the open-loop transfer function $L(s)$ with real part greater than or equal to ϵ , and $N_{ccw}(\epsilon)$ as the number of times the Nyquist curve of $L(\epsilon + j\omega)$ encircles the critical $s = -1$ point in the counterclockwise direction.

Then, the **Nyquist stability criterion** states that if $L(s)$ has no poles with real part equal to ϵ , then $P_{c-l}(\epsilon) = P_{o-l}(\epsilon) - N_{ccw}(\epsilon)$. Thus, $P_{c-l}(\epsilon) = 0$ if and only if $N_{ccw}(\epsilon) = P_{o-l}(\epsilon)$.

A notable special case of this theorem, e.g. **simplified Nyquist stability criterion**, states that if $\epsilon = 0$, i.e. $L(s)$ has no poles on the imaginary axis, then P_{c-l} and P_{o-l} are the number of RHP poles for the closed- and open-loops, respectively (including the imaginary axis) and N_{ccw} is the number of times the Nyquist plot of $L(j\omega)$, i.e. the frequency response of $L(s)$, encircles $s = -1$ in the counterclockwise direction. In this case, the Nyquist theorem states that $P_{c-l} = P_{o-l} - N_{ccw}$ and the closed-loop is stable if and only if $N_{ccw} = P_{o-l}$. Note that this is the case in MATLAB for assessing the Nyquist plot of $L(j\omega)$. Thus, if $L(j\omega)$ has a pole on the imaginary axis, e.g. $K(s)$ has an integral component $\frac{1}{s}$, then care must be taken in the stability analysis.

However, in either case, the Nyquist plot of the open-loop transfer function $L(j\omega)$ provides a visual method for determining the stability of the closed-loop feedback control system, as well as a measure of “how much” stability there is for the system, i.e. its robustness, which will be discussed in the subsequent section.

To prove the general Nyquist theorem, one requires defining a “perturbed” simple, closed curve $\Gamma_{R,\epsilon}$ which is a shifted Γ_R curve by some amount ϵ along the real axis as shown below for some $\epsilon < 0$.



Note that as $\epsilon \rightarrow 0$, $\Gamma_{R,\epsilon} \rightarrow \Gamma_R$. Next, defining the transfer function $H(s) = 1 + L(s)$ and N_p and N_z as the number of poles and zeros of $H(s)$ inside the simple, closed curve $\Gamma_{R,\epsilon}$, then Cauchy's argument principle states that $H(\Gamma_{R,\epsilon})$ has $N_p - N_z$ counterclockwise encirclements of the origin. Then, note the following three observations.

1. As $L(s) = H(s) - 1$, $L(\Gamma_{R,\epsilon})$ is the curve $H(\Gamma_{R,\epsilon})$ shifted to the left by one unit. Thus, $H(\Gamma_{R,\epsilon})$ encircling the origin is equivalent to $L(\Gamma_{R,\epsilon})$ encircling $s = -1$. Furthermore, as $R \rightarrow \infty$ and $\epsilon \rightarrow 0$, $L(\Gamma_{R,\epsilon})$ converges to the Nyquist plot of $L(s)$. Thus, $N_{ccw} = N_p - N_z$.
2. $\Gamma_{R,\epsilon}$ contains the entire RHP as $R \rightarrow \infty$ and $\epsilon \rightarrow 0$. Thus, if R is sufficiently large and ϵ is sufficiently small, then the RHP zeros of $H(s) = 1 + L(s)$ are precisely the closed-loop RHP poles, i.e. $N_z = P_{c-l}$.
3. The RHP poles of $H(s)$ are precisely the RHP poles of the open-loop transfer function $L(s)$, i.e. $N_p = P_{o-l}$. This follows from the assumption that $L(s)$ contains no pole-zero cancellations, and thus, $|H(s_0)| = \infty$ for some s_0 in the RHP if and only if $|L(s_0)| = \infty$.

Combining these three observations, one has that $N_{ccw} = P_{o-l} - P_{c-l}$ or rearranging $P_{c-l} = P_{o-l} - N_{ccw}$ which is what was needed to be proved.

Multivariate Nyquist Criterion

Circle Criterion

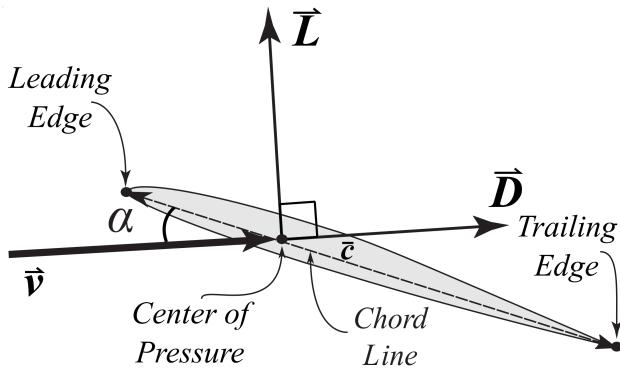
References

Airplane Stability and Control Derivative Models

B.1 Finite-Wing Theory and Component Build-Up Model

Finite-Wing Theory

To develop the basic analytical models for modeling the aerodynamic forces and moments of fixed-wing aircraft, one can use **finite-wing theory** for each lifting surface on a aircraft while also accounting for some interactive aerodynamics between these surfaces and other structures. Finite-wing theory uses the simpler airfoil theory for airflow over any two-dimensional cross-section of a lifting surface, also known as an **airfoil** and resolves the two-dimensional pressure distribution of the moving air over the lifting surface into two perpendicular force contributions as shown in the following free body diagram.



where c is the **aerodynamic chord** from the leading edge to trailing edge, \vec{v}_∞ is the **free-stream velocity**, v_∞ is the **airspeed**, α is the **airfoil angle of attack**, L is the **airfoil lift force** defined as perpendicular to the free-stream velocity and D is the **airfoil drag force** defined as parallel to the free-stream velocity. It should be noted that the **center of pressure** defines the location about which the pitching moment due to

the pressure distribution is currently zero, thus producing no pitching moment. However, as this location will generally change as a function of angle of attack, one often uses the **aerodynamic center** of the airfoil about which the pitching moment will not change with angle of attack, allowing one to resolve the pressure distribution as the lift, drag, and **airfoil pitching moment**, M , at any location.

Considering a finite-wing as a distribution of airfoils at a single angle of attack and defining the **dynamic pressure** for the free-stream velocity, Q_∞ , as

$$Q_\infty = \frac{1}{2} \rho v_\infty^2 \quad (\text{B.1})$$

where ρ is the **air density** which depends on altitude, h , one can model the lift, drag, and wing pitching moment for the entire wing, L_w , D_w , and M_w , respectively, as

$$L_w = Q_\infty S_w C_{L,w} \quad (\text{B.2})$$

$$D_w = Q_\infty S_w C_{D,w} \quad (\text{B.3})$$

and

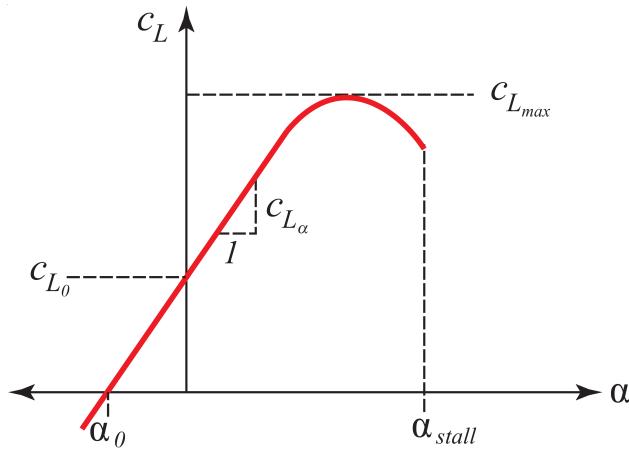
$$M_w = Q_\infty S_w \bar{c}_w C_{M,w} \quad (\text{B.4})$$

where S_w is the surface area of the wing, \bar{c}_w is the **mean aerodynamic chord** of the wing, $C_{L,w}$ is the **wing lift coefficient**, $C_{D,w}$ is the **wing drag coefficient**, and $C_{M,w}$ is the **wing pitching moment coefficient**. Furthermore, due to their finite nature, wings may also have a **rolling moment**, $L_{roll,w}$, defined as

$$L_{roll,w} = Q_\infty S_w b_w C_{L_{roll,w}} \quad (\text{B.5})$$

where b_w is the **wing span** of the wing and $C_{L_{roll,w}}$ is the **wing rolling moment coefficient**. It should be noted that often the “*roll*” subscript is dropped which noticeably overloads the notation for the letter, L , and must be inferred by context whether one is referring to the lift or the rolling moment.

The lift coefficient for a finite-wing generally depends on the wing angle of attack as depicted in the following plot.



At low angles of attack, the relationship is linear, i.e.

$$C_{L,w} = C_{L_\alpha,w}(\alpha_w - \alpha_{0,w}) \quad (\text{B.6})$$

where $C_{L_\alpha,w} = \frac{dC_{L,w}}{d\alpha}$ is the **wing lift-curve slope** and $\alpha_{0,w} \leq 0$ is the **wing zero-lift angle of attack**. This can also be written as

$$C_{L,w} = C_{L_\alpha,w}\alpha_w + C_{L_0,w} \quad (\text{B.7})$$

where $C_{L,0,w}$ is the lift coefficient at $\alpha = 0$ and equals $C_{L,0,w} = -C_{L_\alpha,w}\alpha_{0,w}$. If $C_{L_0,w} = \alpha_{0,w} = 0$, then one has a **symmetric wing**, otherwise one has a **cambered wing**. Note that at high angles of attack, a maximum lift coefficient is reached, $C_{L_{max}}$, after which there is a slight decrease in the lift coefficient and eventually a large decrease. With this large decrease in lift there is also a large increase in drag, a phenomenon known as **stall** which occurs at some α_{stall} . As the purpose of wings are generally to generate lift with a small amount of drag, fixed-wing aircraft nominally operate at angles of attack well below stall and will be assumed in the modeling of this textbook. Stall aerodynamics are beyond the scope of this textbook.

The drag coefficient for a finite-wing flying can generally be separated into three terms as

$$C_{D,w} = C_{D_0,w} + C_{D_i,w} + C_{D_{w,w}} \quad (\text{B.8})$$

where $C_{D_0,w}$ is the **wing parasitic drag coefficient**, also known as the **wing profile drag coefficient**, due to air pressure and skin friction, $C_{D_i,w}$ is the **wing induced drag coefficient** due to the production of lift, and $C_{D_{w,w}}$ is the **wing wave drag coefficient** due to generation of shockwaves at supersonic flight. As $C_{D_0,w}$ is constant with respect to the flight conditions, it is also known as the **zero-lift drag coefficient**.

The induced drag coefficient is typically modeled as a quadratic function

$$C_{D_i,w} = \frac{C_{L,w}^2}{\pi e_0 A R_w} \quad (\text{B.9})$$

where $A R_w$ is the **wing aspect ratio** defined as

$$A R_w = \frac{b_w^2}{S_w} \quad (\text{B.10})$$

and $0 < e_0 < 1$ is the **wing Oswald efficiency** which is typically between 0.7 – 0.85 for subsonic fixed-wing aircraft with moderate aspect ratio and sweep angles. At supersonic speeds, e_0 can range from 0.3 – 0.5. For flat plate airfoils, the wave drag coefficient can be modeled as

$$C_{D_{w,w}} = \frac{4\alpha_w}{\sqrt{M_\infty^2 - 1}} \quad (\text{B.11})$$

where M_∞ is the free-stream Mach number, i.e.

$$M_\infty = \frac{v_\infty}{c_s} \quad (\text{B.12})$$

where c_s is the **speed of sound** of the air mass which varies with the square root of the thermodynamic temperature.

For a wing with constant airfoil shape at subsonic speeds, one can use **lifting-line theory** which predicts that the lift-curve slope for a finite-wing, $C_{L_\alpha, w}$, can be related to the section **airfoil's lift-curve slope**, c_{L_α} , for a wide range of aspect ratios as

$$C_{L_\alpha, w} = \frac{2\pi AR_w}{2 + \sqrt{\frac{4\pi^2 AR_w^2 (1 - M_\infty^2)}{c_{L_\alpha}^2} \left(1 + \frac{\tan^2 \Lambda_{c/2}}{1 - M_\infty^2}\right) + 4}} \text{ rad} \quad (\text{B.13})$$

where $\Lambda_{c/2}$ is the **sweep angle** of the half chord line. In addition, lifting-line theory predicts that the induced drag coefficient can be modeled as

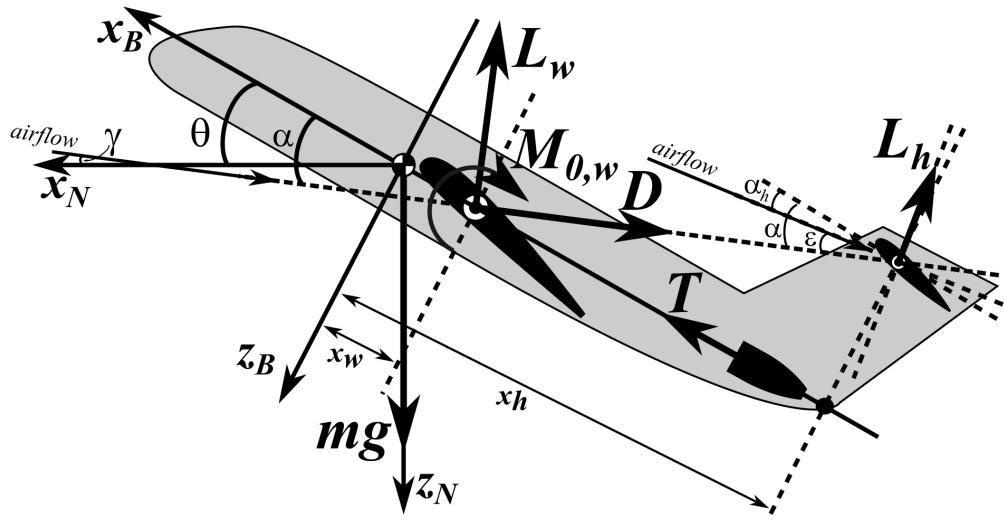
$$C_{D_i, w} = \frac{C_{L_\alpha, w}^2}{\pi AR_w e_w} \quad (\text{B.14})$$

where e_w is the wing's **span efficiency** where if $e_w = 1$, the wing is defined as an **elliptical**. Typically $e_0 < e_w$, but it is often a good approximation without additional data. However, it should be noted that most aircraft use multiple types of airfoils at different points of a lifting surface, a technique called **aerodynamic twisting** which makes an analytical solution more difficult to obtain than lifting-line theory.

Component Build-Up Model

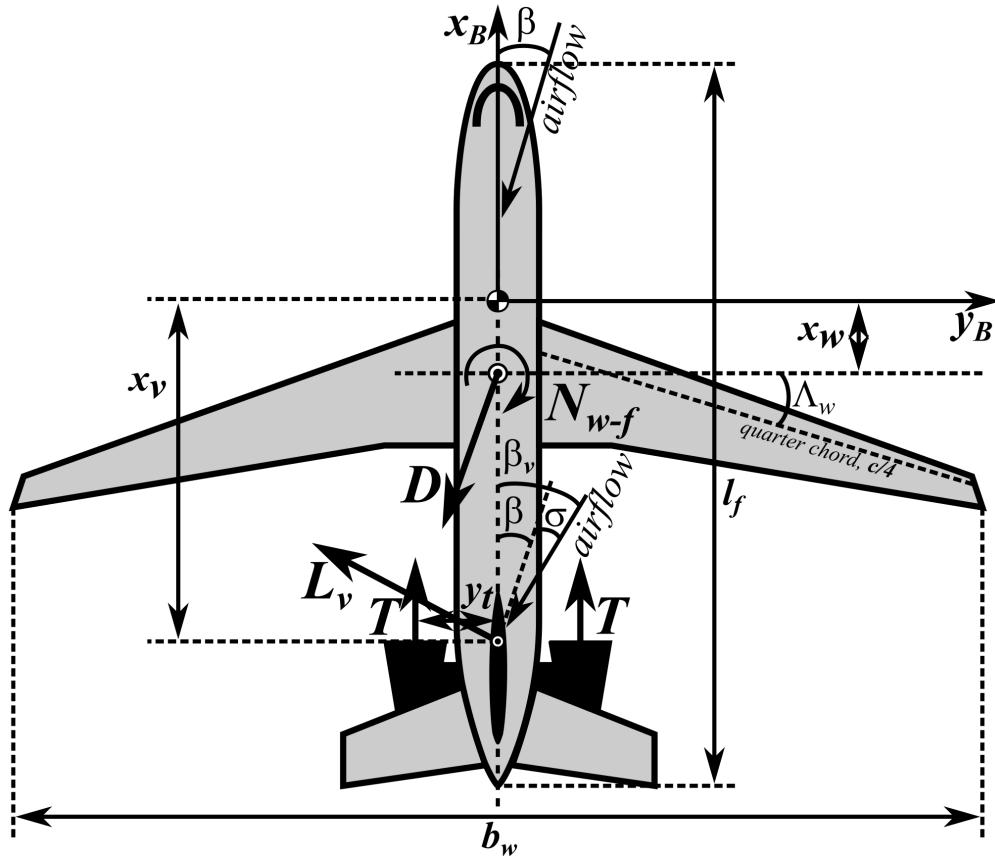
The **component build-up model** uses finite-wing theory for each primary lifting surface of a fixed-wing aircraft to build up the stability and control derivatives. For each surfaces, one must define the aerodynamic center about which lift, drag, pitching moment, and rolling moment can be resolved for that surface as prescribed via finite-wing theory. For conventional fixed-wing aircraft, there are four primary lifting surfaces to consider in aerodynamic coefficient modeling, namely the wing aerodynamic coefficients denoted with subscript w , the horizontal tail aerodynamic coefficients denoted with subscript h , the vertical tail aerodynamic coefficients denoted with subscript v , and the fuselage aerodynamic coefficients denoted with subscript f .

From the view of the $x_B - z_B$ -plane, also known as the **longitudinal plane**, the simplified FBD with the y_B -axis drawn into the page can be drawn as



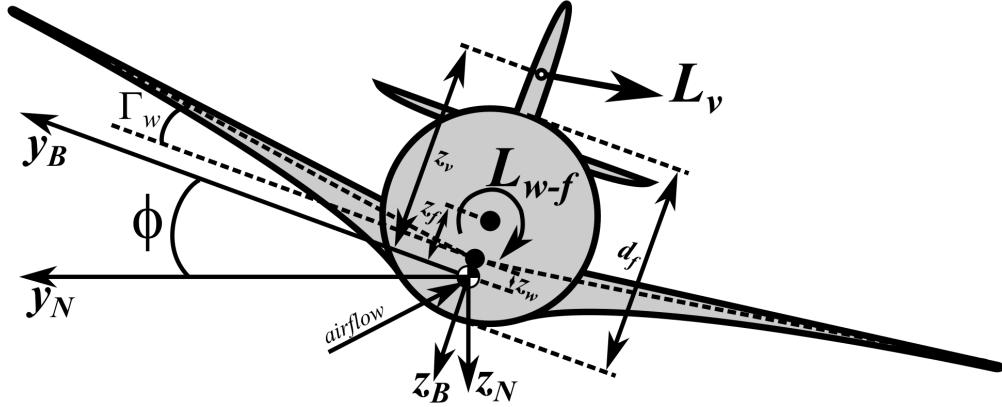
where the airplane's induced drag is dominated by the wing and the parasitic drag has been resolved at the wing (i.e. this replaces D_f , D_w , D_h , and D_v by D at wing aerodynamic center), the drag and thrust contribution to the pitching moment on the airplane has been neglected, the horizontal tail airfoil is symmetric (i.e. $M_{0,h} = 0$), and ϵ is the **downwash angle** which results from the airflow interacting with the wings and engines upstream of the horizontal tail. Note that the aerodynamic centers of each lifting surface in this plane are labeled by their x and z coordinates in the body frame. It should also be noted that the lift from the fuselage is often quite small in this plane.

From the view of the $x_B - y_B$ -plane, also known as the **directional plane**, the simplified FBD with the z_B -axis drawn into the page can be drawn as



where the wing and fuselage effects can be combined into two terms, L_{w-f} and N_{w-f} , the vertical tail airfoil is symmetric (i.e. $N_{0,v} = 0$), and σ is the **sidewash angle** which results from the airflow interacting with the wings and engines upstream of the vertical tail. It should be noted that each thrust vector has a y_B component which cancel each other with respect to the yawing moment.

From the view of the $y_B - z_B$ -plane, also known as the **lateral plane**, the simplified FBD with the x_B -axis drawn out of the page can be drawn as



It should be noted that the aerodynamic centers of the fuselage and wing are additionally represented by z_B coordinates in the body frame which was neglected in the longitudinal frame.

Furthermore, if one assumes the small angle approximation for α and β , $\alpha - \epsilon$, and $\beta + \sigma$, then, one obtains the simplified wind-to-body aerodynamic and propulsive force and moment equations as

$$\begin{aligned}
 mX &= (L_w + L_h)\alpha - D + T \\
 mY &= -D\beta - L_v \\
 mZ &= (L_w + L_h) - D\alpha \\
 I_{xx}L &= L_{w-f} - z_v L_v \\
 I_{yy}M &= M_{0,w} + x_w L_w + x_h L_h \\
 I_{zz}N &= N_{w-f} - x_v L_v
 \end{aligned} \tag{B.15}$$

where α and β are in radians and x_w , x_h , and x_v are the body frame coordinates of the aerodynamic centers, i.e. negative values in the FBD. Note that L and L_{w-f} are moment terms and not lift terms here. By substituting for $(L_w + L_h)$, D , T , L_w , L_h , L_v , and N_v by their aerodynamic coefficients, one has

$$\begin{aligned}
 mX &= Q_\infty S_w C_L \alpha - Q_\infty S_w C_D + Q_\infty S_w C_T \\
 mY &= -Q_\infty S_w C_D \beta - Q_v S_v C_{L,v} \\
 mZ &= -Q_\infty S_w C_L - Q_\infty S_w C_D \alpha \\
 I_{xx}L &= Q_\infty S_w b_w C_{l,w-f} - z_v Q_v S_v C_{L,v} \\
 I_{yy}M &= Q_\infty S_w \bar{c}_w C_{m_0,w} + x_w Q_\infty S_w C_{L,w} + x_h Q_h S_h C_{L,h} \\
 I_{zz}N &= Q_\infty S_w b_w C_{n,w-f} - x_v Q_v S_v C_{L,v}
 \end{aligned} \tag{B.16}$$

where the lowercase l , m , and n are used for the body frame moment coefficient subscripts as C_L is used for the overall airplane lift coefficient, in this case $L_w + L_h$. These equations can be further simplified into

non-dimensional coefficient equations by the definitions

$$\begin{aligned} X &= \frac{Q_\infty S_w}{m} C_X \\ Y &= \frac{Q_\infty S_w}{m} C_Y \\ Z &= \frac{Q_\infty S_w}{m} C_Z \\ L &= \frac{Q_\infty S_w b_w}{I_{xx}} C_l \\ M &= \frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_m \\ N &= \frac{Q_\infty S_w b_w}{I_{zz}} C_n \end{aligned} \quad (\text{B.17})$$

which provides the following approximation for the force and moment coefficients as

$$\begin{aligned} C_X &= C_L \alpha - C_D + C_T \\ C_Y &= -C_D \beta - \frac{Q_v S_v}{Q_\infty S_w} C_{L,v} \\ C_Z &= -C_L - C_D \alpha \\ C_l &= C_{l,w-f} - \frac{Q_v S_v z_v}{Q_\infty S_w b_w} C_{L,v} \\ C_m &= C_{m_0,w} + \frac{x_w}{\bar{c}_w} C_{L,w} - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L,h} \\ C_n &= C_{n,w-f} - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L,v} \end{aligned} \quad (\text{B.18})$$

Note that these simplified equations will also be used in this section to derive analytical models in the rest of this appendix for studying airplane dynamics about steady-flight conditions.

B.2 Longitudinal Stability and Control Derivatives

This section will derive models for the primary longitudinal stability and control derivatives used in the Jacobian linearization of the aerodynamic and propulsive normalized forces and moments about trimmed steady flight, i.e.

$$\begin{aligned} X &= \bar{X} + X_u \Delta u + X_\alpha \Delta \alpha + X_{\delta_T} \Delta \delta_T \\ Z &= \bar{Z} + Z_u \Delta u + Z_\alpha \Delta \alpha + Z_q \Delta q + Z_{\dot{\alpha}} \Delta \dot{\alpha} + Z_{\delta_e} \Delta \delta_e + Z_{\delta_T} \Delta \delta_T \\ M &= \bar{M} + M_u \Delta u + M_\alpha \Delta \alpha + M_q \Delta q + M_{\dot{\alpha}} \Delta \dot{\alpha} + M_{\delta_e} \Delta \delta_e + M_{\delta_T} \Delta \delta_T \end{aligned} \quad (\text{B.19})$$

where the coefficients of the linear terms are called the **longitudinal stability and control derivatives**. Alternatively, these equations can be written in terms of nondimensional coefficients as

$$\begin{aligned} C_X &= \bar{C}_X + C_{X_u} \Delta u + C_{X_\alpha} \Delta \alpha + C_{X_{\delta_T}} \Delta \delta_T \\ C_Z &= \bar{C}_Z + C_{Z_u} \Delta u + C_{Z_\alpha} \Delta \alpha + C_{Z_q} \Delta q + C_{Z_{\dot{\alpha}}} \Delta \dot{\alpha} + C_{Z_{\delta_e}} \Delta \delta_e + C_{Z_{\delta_T}} \Delta \delta_T \\ C_m &= \bar{C}_m + C_{m_u} \Delta u + C_{m_\alpha} \Delta \alpha + C_{m_q} \Delta q + C_{m_{\dot{\alpha}}} \Delta \dot{\alpha} + C_{m_{\delta_e}} \Delta \delta_e + C_{m_{\delta_T}} \Delta \delta_T \end{aligned} \quad (\text{B.20})$$

where the coefficients of the linear terms are called the **longitudinal stability and control coefficients**.

It should be noted that a more comprehensive collection of aerodynamic stability and control prediction techniques can be found in the *USAF Stability and Control DATCOM (Data Compendium)* which was compiled between 1960 and 1978 by the McDonnell Douglas Corporation in conjunction with the Flight Dynamics Laboratory at Wright-Patterson Air Force Base.

Longitudinal Static Stability Coefficient

For the M -moment, the coefficient is

$$C_m = C_{m_0,w} + \frac{x_w}{\bar{c}_w} C_{L,w} - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L,h} \quad (\text{B.21})$$

Next, modeling the lift coefficients for small angles of attack as

$$C_{L,w} = C_{L_0,w} + C_{L_{\alpha,w}} \alpha_w \quad (\text{B.22})$$

and

$$C_{L,h} = C_{L_{\alpha,h}} \alpha_h \quad (\text{B.23})$$

where it has been assumed that the vertical tail has a symmetric airfoil, thus providing no net force at $\alpha_h = 0^\circ$. Then, one has

$$C_m = C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w} + C_{L_{\alpha,w}} \alpha) - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_{\alpha,h}} \alpha_h \quad (\text{B.24})$$

Then, by the previous FBD, one also has

$$\alpha_h = \alpha - \epsilon \quad (\text{B.25})$$

where ϵ is typically approximated by a linear function

$$\epsilon = \frac{d\epsilon}{d\alpha} (\alpha + \alpha_{0,w}) = \frac{d\epsilon}{d\alpha} \alpha + \epsilon_0 \quad (\text{B.26})$$

where for an elliptical lift distribution

$$\epsilon_0 = \frac{2C_{L,w}}{\pi AR_w} \quad (\text{B.27})$$

and

$$\frac{d\epsilon}{d\alpha} = \frac{2C_{L_{\alpha,w}}}{\pi AR_w} \quad (\text{B.28})$$

though these are typically estimated using test data and/or CFD. Next, substituting for α_h and ϵ , one has

$$\begin{aligned} C_m = & C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w} + C_{L_\alpha,w}\alpha) \\ & + \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha,h} \left(\alpha - \left(\epsilon_0 + \frac{d\epsilon}{d\alpha} \alpha \right) \right) \end{aligned} \quad (\text{B.29})$$

and finally grouping constant and α -varying terms provides

$$C_m = \left[C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w} - \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha,h} \epsilon_0) \right] + \left[\frac{x_w}{\bar{c}_w} C_{L_\alpha,w} + \frac{Q_h S_h x_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha,h} \left(1 - \frac{d\epsilon}{d\alpha} \right) \right] \alpha \quad (\text{B.30})$$

and defining

$$\eta_h = \frac{Q_h}{Q_\infty} \quad (\text{B.31})$$

as the **horizontal tail efficiency** and

$$V_h = \frac{-x_h S_h}{\bar{c}_w S_w} \quad (\text{B.32})$$

as the **horizontal tail volume ratio**, one has

$$C_m = C_{m_0} + C_{m_\alpha} \alpha \quad (\text{B.33})$$

where

$$C_{m_0} = C_{m_0,w} + \frac{x_w}{\bar{c}_w} (C_{L_0,w} + \eta_h V_h C_{L_\alpha,h} \epsilon_0) \quad (\text{B.34})$$

and

$$C_{m_\alpha} = \frac{x_w}{\bar{c}_w} C_{L_\alpha,w} - \eta_h V_h C_{L_\alpha,h} \left(1 - \frac{d\epsilon}{d\alpha} \right) \quad (\text{B.35})$$

u Stability Derivatives and Coefficients

For X , consider the dominant drag and thrust forces for steady flight, i.e.

$$mX = -D + T \quad (\text{B.36})$$

or

$$mX = -Q_\infty S_w C_D + T \quad (\text{B.37})$$

and taking the derivative with respect to u

$$mX_u = -Q_\infty S_w \left(\frac{2\bar{C}_D}{u} + C_{D_u} \right) + T_u \quad (\text{B.38})$$

and defining the coefficient

$$X_u = \frac{Q_\infty S_w}{m\bar{u}} C_{X_u} \quad (\text{B.39})$$

one has

$$C_{X_u} = - (2\bar{C}_D + C_{D_u}) + C_{T_u} \quad (\text{B.40})$$

where $C_{T_u} \approx 0$ for jet engines, $C_{T_u} \approx -\bar{C}_D$ for piston engines, and C_{D_u} typically depends on the trim Mach number $\bar{\mathcal{M}}$, i.e.

$$C_{D_u} = C_{D_M} \bar{\mathcal{M}} \quad (\text{B.41})$$

where the Mach number is defined as

$$\bar{\mathcal{M}} = \frac{\bar{v}_\infty}{\bar{v}_s} \quad (\text{B.42})$$

where \bar{v}_s is the speed of sound at trim, and varies with altitude. Also, note that for low speed flight

$$C_{D_u} \approx 0 \quad (\text{B.43})$$

For Z , consider the dominant force

$$mZ = -L \quad (\text{B.44})$$

or by lifting-line theory

$$mZ = -Q_\infty S_w C_L \quad (\text{B.45})$$

and taking the derivative with respect to u , one has

$$mZ_u = -Q_\infty S_w \left(\frac{2\bar{C}_L}{u} + C_{L_u} \right) \quad (\text{B.46})$$

Defining the coefficient as

$$Z_u = \frac{Q_\infty S_w}{m\bar{u}} C_{Z_u} \quad (\text{B.47})$$

one has

$$C_{Z_u} = - (2\bar{C}_L + \bar{u} C_{L_u}) \quad (\text{B.48})$$

The Prantl-Glaudent formula for the lift coefficient of compressible flow given the incompressible lift coefficient, $C_{L,M=0}$, states

$$C_L = \frac{1}{\sqrt{1 - \bar{\mathcal{M}}^2}} C_{L,M=0} \quad (\text{B.49})$$

one can show

$$\bar{u} C_{L_u} = \left(\frac{\bar{\mathcal{M}}^2}{1 - \bar{\mathcal{M}}^2} \right) \bar{C}_L \quad (\text{B.50})$$

Then,

$$C_{Z_u} = - \left(2 + \frac{\bar{\mathcal{M}}^2}{1 - \bar{\mathcal{M}}^2} \right) \bar{C}_L \quad (\text{B.51})$$

For M , the stability coefficient is defined as

$$M_u = \frac{Q_\infty S_w \bar{c}_w}{I_{yy} \bar{u}} C_{m_u} \quad (\text{B.52})$$

for which the stability coefficient can also be rewritten as

$$C_{m_u} = C_{m_M} \bar{\mathcal{M}} \quad (\text{B.53})$$

and for low speed flight

$$C_{m_u} \approx 0 \quad (\text{B.54})$$

α Stability Derivatives and Coefficients

For X , consider the coefficient form where one has

$$C_X = C_L\alpha - C_D + C_T \quad (\text{B.55})$$

thus, the stability coefficient with respect to α is

$$C_{X_\alpha} = \bar{C}_L - C_{D_\alpha} \quad (\text{B.56})$$

where, by lifting-line theory, one has

$$C_{D_\alpha} = \frac{2\bar{C}_L C_{L_\alpha}}{\pi A R_w e_w} \quad (\text{B.57})$$

where the stability derivative is defined as

$$X_\alpha = \frac{Q S_w}{m} C_{X_\alpha} \quad (\text{B.58})$$

For Z , consider

$$C_Z = -(C_L + C_D\alpha) \quad (\text{B.59})$$

$$C_Z = -(C_{L_\alpha}\alpha + C_{L_0} + C_D\alpha) \quad (\text{B.60})$$

$$C_Z = -(C_{L_\alpha} + C_D)\alpha - C_{L_0} \quad (\text{B.61})$$

thus, the stability coefficient with respect to α is

$$C_{Z_\alpha} = -(C_{L_\alpha} + \bar{C}_D) \quad (\text{B.62})$$

where the stability derivative is defined as

$$Z_\alpha = \frac{Q_\infty S_w}{m} C_{Z_\alpha} \quad (\text{B.63})$$

For M , the stability derivative is defined as

$$M_\alpha = \frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_\alpha} \quad (\text{B.64})$$

where the stability coefficient was derived previously for static stability as

$$C_{m_\alpha} = \frac{x_w}{\bar{c}_w} C_{L_{\alpha,w}} - \eta_h V_h C_{L_{\alpha,h}} \left(1 - \frac{d\epsilon}{d\alpha} \right) \quad (\text{B.65})$$

It should be noted that sometimes a $C_{m_{\alpha,f}}$ term is included in this derivative to account for fuselage effects.

q Stability Derivatives and Coefficients

For Z , the primary change is due to the change of the lift at the horizontal tail. Thus, consider modeling the change in tail lift as

$$\Delta L_h = Q_h S_h C_{L_\alpha, h} \Delta \alpha_h \quad (\text{B.66})$$

where due to a rotation rate of q provides by small angle approximation

$$\Delta \alpha_h = -\frac{x_h q}{\bar{u}} \quad (\text{B.67})$$

Thus,

$$\Delta L_h = -Q_h S_h C_{L_\alpha, h} \frac{x_h q}{\bar{u}} \quad (\text{B.68})$$

or in terms of the stability derivative where L_h acts opposite the z_B axis

$$m \Delta Z = Q_h S_h C_{L_\alpha, h} \frac{x_h q}{\bar{u}} \quad (\text{B.69})$$

and taking the derivative with respect to q

$$Z_q = Q_h S_h C_{L_\alpha, h} \frac{x_h}{m \bar{u}} \quad (\text{B.70})$$

By definition of the stability derivative as

$$Z_q = \frac{Q_\infty S_w \bar{c}_w}{2m \bar{u}} C_{Z_q} \quad (\text{B.71})$$

the stability coefficient is

$$C_{Z_q} = \frac{2Q_h x_h S_h}{Q_\infty S_w \bar{c}_w} C_{L_\alpha, h} \quad (\text{B.72})$$

Then, defining

$$V_h = \frac{-x_h S_h}{S_w \bar{c}_w} \quad (\text{B.73})$$

as the **horizontal tail volume ratio** and

$$\eta_h = \frac{Q_h}{Q_\infty} \quad (\text{B.74})$$

as the **horizontal tail efficiency**, one has

$$C_{Z_q} = -2\eta_h V_h C_{L_\alpha, h} \quad (\text{B.75})$$

Note that x_h is a negative number here. It should also be noted that typical values for η_h are 0.8-1.2 where η_h can be > 1 because of slipstream or engine stream effects, i.e. the interaction of the airflow with the wing or engine before encountering the horizontal tail.

For M , consider the moment due to Z_q as modeled by the horizontal tail, i.e.

$$I_{yy} M_q = -x_h m Z_q \quad (\text{B.76})$$

By definition of the stability derivative as

$$M_q = \frac{Q_\infty S_w \bar{c}_w^2}{2I_{yy}\bar{u}} C_{m_q} \quad (\text{B.77})$$

and substituting for Z_q from the previous derivation, the stability coefficient is

$$C_{m_q} = -\frac{x_h}{\bar{c}_w} C_{Z_q} \quad (\text{B.78})$$

Note that Z_q was neglected in the linearized EOMs in the previous lecture due to its small magnitude relative to \bar{u} , though it appears indirectly in M_q .

Lastly, note that a common practice with these q derivatives is to increase these models by 10% to approximately account for the wing and fuselage as well.

$\dot{\alpha}$ Stability Derivatives and Coefficients

For Z , consider the change in the angle of attack primarily changing the circulation around the wing. This most directly affects the downwash at the tail which occurs at a lag time approximated by

$$\Delta t = \frac{-x_t}{\bar{u}} \quad (\text{B.79})$$

Furthermore, the change in downwash can be related to the change in the horizontal tail angle of attack as

$$\Delta\alpha_h = \frac{d\epsilon}{dt} \Delta t \quad (\text{B.80})$$

Then, by substitution

$$\Delta\alpha_h = \frac{d\epsilon}{dt} \frac{-x_h}{\bar{u}} \quad (\text{B.81})$$

$$\Delta\alpha_h = \frac{d\epsilon}{d\alpha} \frac{d\alpha}{dt} \frac{-x_h}{\bar{u}} \quad (\text{B.82})$$

$$\Delta\alpha_h = \frac{d\epsilon}{d\alpha} \dot{\alpha} \frac{-x_h}{\bar{u}} \quad (\text{B.83})$$

and relating this to the change in the lift, one has

$$\Delta L_h = Q_h S_h C_{L_{\alpha},h} \Delta\alpha_h \quad (\text{B.84})$$

or in terms of Z where L_h acts opposite the z_B axis

$$m\Delta Z = -Q_h S_h C_{L_{\alpha},h} \frac{d\epsilon}{d\alpha} \dot{\alpha} \frac{-x_h}{\bar{u}} \quad (\text{B.85})$$

and taking the derivative with respect to $\dot{\alpha}$

$$Z_{\dot{\alpha}} = -Q_h S_h \frac{-x_h}{m\bar{u}} C_{L_{\alpha},h} \frac{d\epsilon}{d\alpha} \quad (\text{B.86})$$

By definition of the stability derivative as

$$Z_{\dot{\alpha}} = \frac{Q_{\infty} S_w \bar{c}_w}{2m\bar{u}} C_{Z_{\dot{\alpha}}} \quad (\text{B.87})$$

the stability coefficient is

$$C_{Z_{\dot{\alpha}}} = -2V_h \eta_h C_{L_{\alpha},h} \frac{d\epsilon}{d\alpha} \quad (\text{B.88})$$

For M , consider the moment due to $Z_{\dot{\alpha}}$ as modeled by the horizontal tail, i.e.

$$I_{yy} M_{\dot{\alpha}} = -x_h m Z_{\dot{\alpha}} \quad (\text{B.89})$$

By definition of the stability derivative as

$$M_{\dot{\alpha}} = \frac{Q_{\infty} S_w \bar{c}_w^2}{2I_{yy}\bar{u}} C_{m_{\dot{\alpha}}} \quad (\text{B.90})$$

and substituting for $Z_{\dot{\alpha}}$ from the previous derivation, the stability coefficient is

$$C_{m_{\dot{\alpha}}} = -\frac{x_h}{\bar{c}_w} C_{Z_{\dot{\alpha}}} \quad (\text{B.91})$$

Note that $Z_{\dot{\alpha}}$ was neglected in the linearized EOMs in the previous lecture due to its small magnitude relative to \bar{u} , although it appears indirectly in $M_{\dot{\alpha}}$

δ_T and δ_e Control Derivatives and Coefficients

The longitudinal control inputs are the elevator angle, δ_e , and throttle input δ_T . The throttle coefficients and derivatives are determined solely by the engine type and can vary with the placement of the engines, e.g. under the wings. However, their modeling is beyond the scope of this course, but, in general the thrust can affect C_X , C_Z , and C_m . This implies the following definitions for the control derivatives

$$X_{\delta_T} = \frac{Q_{\infty} S_w}{m} C_{X_{\delta_T}} \quad (\text{B.92})$$

$$Z_{\delta_T} = \frac{Q_{\infty} S_w}{m} C_{Z_{\delta_T}} \quad (\text{B.93})$$

$$M_{\delta_T} = \frac{Q_{\infty} S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_T}} \quad (\text{B.94})$$

where the coefficients will depend on the design of the propulsion system.

Recalling for airplanes, one typically models the propulsive force as

$$\vec{F}_{p,B} = \begin{bmatrix} T \cos \theta_T \\ 0 \\ T \sin \theta_T \end{bmatrix} \quad (\text{B.95})$$

where T is the **thrust force** and θ_T is a potential offset angle with respect to the x_B -axis of the body-fixed frame. Moreover, a propulsive moment may also be present nominally about the y_B -axis as

$$\vec{M}_{p,B} = \begin{bmatrix} 0 \\ T(z_T \cos \theta_T - x_T \sin \theta_T) \\ 0 \end{bmatrix} \quad (\text{B.96})$$

where (x_T, z_T) denotes the location of the thrust force in the $x_B - z_B$ plane. It should be noted that often $\theta_T = 0^\circ$ or approximately and T is generally a function of the airspeed, altitude, and throttle setting. And modeling the thrust force as a **thrust coefficient**, i.e.

$$T = Q_\infty S_w C_T \quad (\text{B.97})$$

one has

$$C_{X_{\delta_T}} = \cos \theta_T C_T \quad (\text{B.98})$$

$$C_{Z_{\delta_T}} = \sin \theta_T C_T \quad (\text{B.99})$$

$$C_{m_{\delta_T}} = (z_T \cos \theta_T - x_T \sin \theta_T) C_T \quad (\text{B.100})$$

The elevator angle input affects the aerodynamics of the horizontal tail which will affect the lift and induced drag of the tail, the second of which is relatively small compared to the total airplane drag and will be neglected in this course's models. Thus, the elevator control derivatives are defined as

$$Z_{\delta_e} = \frac{Q_\infty S_w}{m} C_{Z_{\delta_e}} \quad (\text{B.101})$$

$$M_{\delta_e} = \frac{Q_\infty S_w \bar{c}_w}{I_{yy}} C_{m_{\delta_e}} \quad (\text{B.102})$$

where the coefficients can be modeled using the tail lift effects as

$$C_{Z_{\delta_e}} = -C_{L_{\delta_e}} = -\eta_h \frac{S_h}{S_w} C_{L_{\delta_e}, h} \quad (\text{B.103})$$

$$C_{m_{\delta_e}} = -\frac{x_h}{\bar{c}_w} C_{Z_{\delta_e}} \quad (\text{B.104})$$

$C_{m_{\delta_e}}$ is also known as the **elevator control power** while the derivative $C_{L_{\delta_e}}$ is the **elevator effectiveness** and can be rewritten as

$$C_{L_{\delta_e}} = \eta_h \frac{S_h}{S_w} \frac{\partial C_{L,h}}{\partial \delta_e} = \eta_h \frac{S_h}{S_w} \frac{\partial C_{L,h}}{\partial \alpha_h} \frac{\partial \alpha_h}{\partial \delta_e} = \eta_h \frac{S_h}{S_w} C_{L_{\alpha,h}} \tau_e \quad (\text{B.105})$$

where the τ **empirical parameter** models the relationship between the change in the lifting surface's angle of attack and the control surface deflection angle. Intuitively, τ depends on the ratio of the control surface area to the lifting surface area, e.g. if S_e is the surface area of the elevator, then τ_e is related to S_e/S_h . This term will also be used for the rudder and aileron control coefficients. The following table provides discrete values for τ .

| | | | | | | | | |
|---------------------------|------|------|------|------|------|------|------|------|
| $S_{control}/S_{lifting}$ | 0.0 | 0.05 | 0.10 | 0.15 | 0.20 | 0.25 | 0.30 | 0.35 |
| τ | 0.0 | 0.16 | 0.26 | 0.34 | 0.41 | 0.47 | 0.52 | 0.56 |
| $S_{control}/S_{lifting}$ | 0.40 | 0.45 | 0.50 | 0.55 | 0.60 | 0.65 | 0.70 | |
| τ | 0.60 | 0.64 | 0.68 | 0.72 | 0.75 | 0.78 | 0.80 | |

For intermediate values one typically uses linear interpolation.

B.3 Lateral-Directional Stability and Control Derivatives

This section will derive models for the lateral stability and control derivatives used in the Jacobian linearization of the normalized aerodynamic forces and moments about trimmed steady flight, i.e.

$$\begin{aligned} Y &= \bar{Y} + Y_\beta \Delta\beta + Y_p \Delta p + Y_r \Delta r + Y_{\delta_r} \Delta\delta_r \\ L &= \bar{L} + L_\beta \Delta\beta + L_p \Delta p + L_r \Delta r + L_{\delta_a} \Delta\delta_a + L_{\delta_r} \Delta\delta_r \\ N &= \bar{N} + N_\beta \Delta\beta + N_p \Delta p + N_r \Delta r + N_{\delta_a} \Delta\delta_a + N_{\delta_r} \Delta\delta_r \end{aligned} \quad (\text{B.106})$$

where the coefficients of the linear terms are called the **lateral-directional stability and control derivatives**. Alternatively, these equations can be written in terms of nondimensional coefficients as

$$\begin{aligned} C_Y &= \bar{C}_Y + C_{Y_\beta} \Delta\beta + C_{Y_p} \Delta p + C_{Y_r} \Delta r + C_{Y_{\delta_r}} \Delta\delta_r \\ C_l &= \bar{C}_l + C_{l_\beta} \Delta\beta + C_{l_p} \Delta p + C_{l_r} \Delta r + C_{l_{\delta_a}} \Delta\delta_a + C_{l_{\delta_r}} \Delta\delta_r \\ C_n &= \bar{C}_n + C_{n_\beta} \Delta\beta + C_{n_p} \Delta p + C_{n_r} \Delta r + C_{n_{\delta_a}} \Delta\delta_a + C_{n_{\delta_r}} \Delta\delta_r \end{aligned} \quad (\text{B.107})$$

where the coefficients of the linear terms are called the **lateral-directional stability and control coefficients**.

It should be noted that a more comprehensive collection of aerodynamic stability and control prediction techniques can be found in the *USAF Stability and Control DATCOM (Data Compendium)* which was compiled between 1960 and 1978 by the McDonnell Douglas Corporation in conjunction with the Flight Dynamics Laboratory at Wright-Patterson Air Force Base.

Lastly, it should also be noted that in the following models S_v is the vertical tail area and includes the submerged area to fuselage centerline.

Lateral-Directional Static Stability Coefficients

For L and N moments, the coefficients are

$$C_l = C_{l,w-f} - \frac{Q_v S_v z_v}{Q_\infty S_w b_w} C_{L,v} \quad (\text{B.108})$$

and

$$C_n = C_{n,w-f} - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L,v} \quad (\text{B.109})$$

Next, modeling the moment and lift coefficients for small angles of attack as

$$C_{l,w-f} = C_{l_\beta,w-f} \beta \quad (\text{B.110})$$

$$C_{n,w-f} = C_{n_\beta,w-f}\beta \quad (\text{B.111})$$

and

$$C_{L,v} = C_{L_\alpha,v}\alpha_v \quad (\text{B.112})$$

where these surfaces are symmetric with respect to β and α_v , respectively, thus providing no net moment or force at 0° . Then, one has

$$C_l = C_{l_\beta,w-f}\beta - \frac{Q_v S_v z_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \alpha_v \quad (\text{B.113})$$

and

$$C_n = C_{n_\beta,w-f}\beta - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \alpha_v \quad (\text{B.114})$$

Then, inspecting the FBD above, one can see that

$$\alpha_v = \beta + \sigma \quad (\text{B.115})$$

where σ can be approximated by a linear function

$$\sigma = \frac{d\sigma}{d\beta} \beta \quad (\text{B.116})$$

which is typically estimated using test data and/or CFD. Next, substituting for α_v and σ , one has

$$C_l = C_{l_\beta,w-f}\beta - \frac{Q_v S_v z_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \left(\beta + \frac{d\sigma}{d\beta} \beta \right) \quad (\text{B.117})$$

and

$$C_n = C_{n_\beta,w-f}\beta - \frac{Q_v S_v x_v}{Q_\infty S_w b_w} C_{L_\alpha,v} \left(\beta + \frac{d\sigma}{d\beta} \beta \right) \quad (\text{B.118})$$

Finally, defining

$$\eta_v = \frac{Q_v}{Q_\infty} \quad (\text{B.119})$$

as the **vertical tail efficiency** and

$$V_v = \frac{-x_v S_v}{b_w S_w} \quad (\text{B.120})$$

as the **vertical tail volume ratio**, one has

$$C_l = C_{l_\beta}\beta \quad (\text{B.121})$$

$$C_n = C_{n_\beta}\beta \quad (\text{B.122})$$

where

$$C_{l_\beta} = C_{l_\beta,w-f} - \eta_v \frac{S_v z_v}{S_w b_w} C_{L_\alpha,v} \left(1 + \frac{d\sigma}{d\beta} \right) \quad (\text{B.123})$$

and

$$C_{n_\beta} = C_{n_\beta,w-f} + \eta_v V_v C_{L_\alpha,v} \left(1 + \frac{d\sigma}{d\beta} \right) \quad (\text{B.124})$$

β Stability Derivatives and Coefficients

For Y , consider the dominant force from the vertical tail

$$mY = -L_v \quad (\text{B.125})$$

and in coefficient form

$$mY = -Q_v S_v C_{L,v} \quad (\text{B.126})$$

or

$$mY = -Q_v S_v C_{L_{\alpha},v} (\beta + \sigma) \quad (\text{B.127})$$

and taking the derivative with respect to β provides

$$mY_{\beta} = -Q_v S_v C_{L_{\alpha},v} \left(1 + \frac{d\sigma}{d\beta} \right) \quad (\text{B.128})$$

By definition of the stability derivative as

$$Y_{\beta} = \frac{Q_{\infty} S_w}{m} C_{Y_{\beta}} \quad (\text{B.129})$$

and defining

$$\eta_v = \frac{Q_v}{Q_{\infty}} \quad (\text{B.130})$$

as the **vertical tail efficiency**, the stability coefficient is

$$C_{Y_{\beta}} = -\eta_v \frac{S_v}{S_w} C_{L_{\alpha},v} \left(1 + \frac{d\sigma}{d\beta} \right) \quad (\text{B.131})$$

The following empirical equation can be used in the previous equation

$$\eta_v \left(1 + \frac{d\sigma}{d\beta} \right) = 0.724 + 3.06 \frac{S_v/S_w}{1 + \cos \Lambda_w} + 0.4 \frac{z_w - z_f}{d_f} + 0.009 AR_w \quad (\text{B.132})$$

where Λ_w is the sweep angle of the wing (measured at quarter chord), z_w is the z_B coordinate of the aerodynamic center of the wing, z_f is the z_B coordinate of the fuselage centerline, d_f is the maximum fuselage depth, and AR_w is the wing aspect ratio.

For L , the primary factors are the dihedral angle and the wing taper. The stability derivative is defined as

$$L_{\beta} = \frac{Q_{\infty} S_w b_w}{I_{xx}} C_{l_{\beta}} \quad (\text{B.133})$$

and the stability coefficient can be simply modeled by

$$C_{l_{\beta}} = \frac{\partial C_{l_{\beta}}}{\partial \Gamma_w} \Gamma_w \quad (\text{B.134})$$

where Γ_w is the wing dihedral angle, $\frac{\partial C_{l_{\beta}}}{\partial \Gamma_w}$ is related to the aspect ratio and taper ratio of the root chord and tip chord (≈ -0.4 to -0.9 /rad²).

For N , the stability derivative is defined as

$$N_\beta = \frac{Q_\infty S_w b_w}{I_{zz}} C_{n_\beta} \quad (\text{B.135})$$

where the stability coefficient was derived previously for static stability as

$$C_{n_\beta} = C_{n_\beta, w-f} + \eta_v V_v C_{L_{\alpha v}} \left(1 + \frac{d\sigma}{d\beta} \right) \quad (\text{B.136})$$

where

$$V_v = \frac{-x_v S_v}{b_w S_w} \quad (\text{B.137})$$

is the **vertical tail volume ratio**.

p Stability Derivatives and Coefficients

For Y , there will be a resulting side force only if there is wing sweep as the change in the relative airflow will only then have a component in the y_B direction. The stability derivative is defined as

$$Y_p = \frac{Q_\infty S_w b_w}{2m\bar{u}} C_{Y_p} \quad (\text{B.138})$$

It can be shown that the stability coefficient is

$$C_{Y_p} = \frac{AR_w + \cos \Lambda_w}{AR_w + 4 \cos \Lambda_w} \tan \Lambda_w \bar{C}_L \quad (\text{B.139})$$

For L , the primary factor is the change in lift distribution over the wing. Thus, consider modeling the change in tail lift over a cross-section of the wing of width dy and chord length $c_w(y)$ as

$$\Delta(\text{Lift}) = Q_\infty(c_w(y)dy)C_{l_{\alpha,w}}\Delta\alpha \quad (\text{B.140})$$

where $C_{l_{\alpha,w}}$ is the cross-sectional lift coefficient slope. A rotation rate of p provides by the small angle approximation

$$\Delta\alpha = \frac{yp}{\bar{u}} \quad (\text{B.141})$$

Thus, for the moment arm y for the contribution to the normalized moment L of the sectional mass

$$my^2\Delta L = -Q_\infty C_{l_{\alpha,w}} \frac{p}{\bar{u}} c_w(y) y^2 dy \quad (\text{B.142})$$

which can be integrated over half the span of the wing twice for the entire roll moment $I_{xx}L$ due to the moment being in the opposite sense for the other side of the wing as

$$I_{xx}L = -2 \int_0^{b_w/2} Q_\infty C_{l_{\alpha,w}} \frac{p}{\bar{u}} c_w(y) y^2 dy \quad (\text{B.143})$$

Then, assuming that the cross-sectional lift coefficient slope is well approximated by the wing lift coefficient slope, one has

$$I_{xx}L = \frac{-2Q_\infty C_{L_{\alpha,w}} p}{\bar{u}} \int_0^{b_w/2} c_w(y) y^2 dy \quad (\text{B.144})$$

and taking the derivative with respect to p

$$I_{xx}L_p = \frac{-2Q_\infty C_{L_{\alpha,w}}}{\bar{u}} \int_0^{b_w/2} c_w(y) y^2 dy \quad (\text{B.145})$$

By definition of the stability derivative as

$$L_p = \frac{Q_\infty S_w b_w^2}{2I_{xx}\bar{u}} C_{l_p} \quad (\text{B.146})$$

The stability coefficient is

$$C_{l_p} = -\frac{4C_{L_{\alpha,w}}}{S_w b_w^2} \int_0^{b_w/2} c_w(y) y^2 dy \quad (\text{B.147})$$

where $c_w(y)$ is the wing chord as a function of the lateral coordinate. For a tapered wing this chord function can be written as

$$c_w(y) = c_{r,w} \left[1 + \left(\frac{\frac{c_{t,w}}{c_{r,w}} - 1}{\frac{b_w}{2}} \right) y \right] \quad (\text{B.148})$$

where $c_{r,w}$ and $c_{t,w}$ are the root and tip chords for the wing which results in a stability coefficient

$$C_{l_p} = -\left(\frac{1+3\lambda_w}{1+\lambda_w} \right) \frac{C_{L_{\alpha,w}}}{12} \quad (\text{B.149})$$

where λ_w is the taper ratio of the wing, i.e.

$$\lambda_w = \frac{c_{t,w}}{c_{r,w}} \quad (\text{B.150})$$

For N , the stability derivative is defined as

$$N_p = \frac{Q_\infty S_w b_w^2}{2I_{zz}\bar{u}} C_{n_p} \quad (\text{B.151})$$

It can be shown that the stability coefficient is

$$C_{n_p} = -\frac{\bar{C}_L}{8} \quad (\text{B.152})$$

which is due to conservation of angular momentum.

r Stability Derivatives and Coefficients

For Y , consider the dominant force from the vertical tail side force and its moment arm due to the resulting sideslip, i.e.

$$Y_r = \frac{x_v}{\bar{u}} Y_{\beta,v} \quad (\text{B.153})$$

$$Y_r = \frac{x_v Q_\infty S_w}{m \bar{u}} C_{Y_{\beta,v}} \quad (\text{B.154})$$

By definition of the stability derivative as

$$Y_r = \frac{Q_\infty S_w b_w}{2m \bar{u}} C_{Y_r} \quad (\text{B.155})$$

the stability coefficient is

$$C_{Y_r} = 2 \frac{x_v}{b_w} C_{Y_{\beta,v}} \quad (\text{B.156})$$

For L , the stability derivative is defined as

$$L_r = \frac{Q_\infty S_w b_w^2}{2I_{xx} \bar{u}} C_{l_r} \quad (\text{B.157})$$

It can be shown that the stability coefficient is

$$C_{l_r} = \frac{\bar{C}_L}{4} - 2 \frac{x_v z_v}{b_w^2} C_{Y_{\beta,v}} \quad (\text{B.158})$$

For N , consider the dominant force from the vertical tail side force and its moment arm, i.e.

$$I_{zz} \Delta N = -Q_v S_v C_{L_{\alpha,v}} x_v \Delta \beta \quad (\text{B.159})$$

where due to a rotation rate of r provides by small angle approximation

$$\Delta \beta = \frac{x_v r}{\bar{u}} \quad (\text{B.160})$$

Thus,

$$I_{zz} \Delta N = -Q_v S_v C_{L_{\alpha,v}} \frac{x_v^2 r}{\bar{u}} \quad (\text{B.161})$$

and taking the derivative with respect to r

$$I_{zz} N_r = -Q_v S_v C_{L_{\alpha,v}} \frac{x_v^2}{\bar{u}} \quad (\text{B.162})$$

By definition of the stability derivative as

$$N_r = \frac{Q_\infty S_w b_w^2}{2I_{zz} \bar{u}} C_{n_r} \quad (\text{B.163})$$

the stability coefficient is

$$C_{n_r} = 2 \eta_v V_v \frac{x_v}{b_w} C_{L_{\alpha,v}} \quad (\text{B.164})$$

Note that $C_{Y_{\beta,v}} = C_{Y_{\beta}}$ for the tail-only model and x_v is a negative number in this appendix.

δ_a and δ_r Control Derivatives and Coefficients

The lateral-directional control derivatives are defined as

$$Y_{\delta_r} = \frac{Q_\infty S_w}{m} C_{Y_{\delta_r}} \quad (\text{B.165})$$

$$L_{\delta_a} = \frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\delta_a}} \quad (\text{B.166})$$

$$L_{\delta_r} = \frac{Q_\infty S_w b_w}{I_{xx}} C_{l_{\delta_r}} \quad (\text{B.167})$$

$$N_{\delta_a} = \frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\delta_a}} \quad (\text{B.168})$$

$$N_{\delta_r} = \frac{Q_\infty S_w b_w}{I_{zz}} C_{n_{\delta_r}} \quad (\text{B.169})$$

For low speed flight, the following models can be used for the control coefficients.

The δ_a control coefficient for L can be modeled as

$$C_{l_{\delta_a}} = \frac{2C_{L_{\alpha,w}}\tau_a}{S_w b_w} \int_{y_{a,i}}^{y_{a,o}} c_w(y) y dy \quad (\text{B.170})$$

where $C_{l_{\delta_a}}$ is the **aileron control power**, $C_{L_{\alpha,w}}\tau_a$ is the **aileron effectiveness**, $y_{a,i}$ is the inner aileron y_B coordinate, and $y_{a,o}$ is the outer aileron y_B coordinate. For a tapered wing, this becomes

$$C_{l_{\delta_a}} = \frac{2C_{L_{\alpha,w}}\tau_a}{S_w b_w} \int_{y_{a,i}}^{y_{a,o}} c_{r,w} \left[1 + \left(\frac{c_{t,w}/c_{r,w} - 1}{b_w/2} \right) y \right] y dy \quad (\text{B.171})$$

$$C_{l_{\delta_a}} = \frac{2C_{L_{\alpha,w}}\tau_a c_{r,w}}{S_w b_w} \left[\frac{y^2}{2} + \left(\frac{c_{t,w}/c_{r,w} - 1}{b_w/2} \right) \frac{y^3}{3} \right]_{y_{a,i}}^{y_{a,o}} \quad (\text{B.172})$$

The δ_a control coefficient for N can be modeled as

$$C_{n_{\delta_a}} = 2K \bar{C}_L C_{l_{\delta_a}} \quad (\text{B.173})$$

where K is an aileron empirical factor (≈ -0.3 to -0.1) that decreases with aspect ratio, increases slightly with aileron placement further out on the wing, and increases with taper ratio; and

The δ_r control coefficient for Y can be modeled as

$$C_{Y_{\delta_r}} = \frac{S_v}{S_w} \tau_r C_{L_{\alpha,v}} \quad (\text{B.174})$$

The δ_r control coefficient for L can be modeled as

$$C_{l_{\delta_r}} = \frac{S_v}{S_w} \tau_r \frac{z_v}{b_w} C_{L_{\alpha,v}} \quad (\text{B.175})$$

The δ_r control coefficient for N can be modeled as

$$C_{n_{\delta_r}} = -\eta_v V_v C_{L_{\alpha,v}} \tau_r \quad (\text{B.176})$$

where $C_{n_{\delta_r}}$ is the **rudder control power** and $C_{L_{\alpha,v}}\tau_r$ is the **rudder effectiveness**.



Bibliography

- [1] R. Nelson, *Flight Stability and Automatic Control*. McGraw-Hill Companies, Inc., 2nd ed., 1998.
- [2] D. K. Schmidt, *Modern Flight Dynamics*. McGraw-Hill Companies, Inc., 1st ed., 2012.
- [3] B. Stevens, F. Lewis, and E. Johnson, *Aircraft Control and Simulation*. John Wiley & Sons, Inc., 3rd ed., 2016.
- [4] V. Klein and E. A. Morelli, *Aircraft System Identification: Theory and Practice*. American Institute of Aeronautics and Astronautics, 2006.
- [5] E. Lavretsky and K. Wise, *Robust and Adaptive Control with Aerospace Applications*. Springer-Verlag London, 2013.
- [6] D. Simon, *Optimal State Estimation: Kalman, H_∞ , and Nonlinear Approaches*. Springer Science & Business Media, 2006.
- [7] R. Van Der Merwe and E. Wan, *Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models*. PhD thesis, Oregon University, 2004.
- [8] G. Gullerud and F. Paganini, *A Course in Robust Control Theory: A Convex Approach*. Springer Science & Business Media, 1st ed., 2000.
- [9] S. Gleason and D. Gebre-Egziabher, *GNSS Applications and Methods*. Artech House, 2009.
- [10] P. Groves, *Principles of GNSS, Inertial, and Multisensor Integrated Navigation Systems*. Artech House, 2nd ed., 2013.
- [11] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Artech House, 1st ed., 2007.

Index

- J_2
WGS 84, 298
- L_2 -norm, 741
weighted, 741
- L_∞ -norm, 741
- L_p -norms, 741
- $L_{2,2}$ -norm, 742
- $L_{\infty,\infty}$ -norm, 742
- M -sample variance, 617
- N -model redundancy, 609
- $\alpha - \beta$ filter, 178
- χ^2 -test, 443
- δ -generalized labeled multi-Bernoulli random finite set, 733
- \forall , 9
- γ -iteration, 157
- \mathcal{H}_∞ loop-shaping
mixed-sensitivity, 159
- \mathcal{L}_1 -norm
signal, 101
- \mathcal{L}_2 -norm
signal, 101
- \mathcal{L}_p -norm
signal, 101
- $\mathcal{L}_{p,q}$ -induced norm
system, 102
- σ -additivity, 403
- σ -plot, 100
- τ empirical parameter, 769
table, 769
- k -nearest-neighbors association, 685
- p -norm, 741
signal, 101
- t -test, 442
- \therefore , 9
- a posteriori* PDF, 472
- a priori* PDF, 472
- 2-sample variance, 617
- A-tail, 214
- acceleration
centrifugal three-dimensional, 229
centrifugal two-dimensional, 226
Coriolis three-dimensional, 229
Coriolis two-dimensional, 226
Euler three-dimensional, 229
Euler two-dimensional, 226
fictitious, 226
two-dimensional, 225
- acceleration model
constant-acceleration, 701
constant-velocity, 700
Singer, 701
white noise, 699

- Wiener process, 700
- Wiener sequence, 701
- acceleration-to-speed ratio, 717
- accelerometer, 583, 621
- acceptance probability, 569
- acceptance-rejection sampling, 419
- active
 - energy sensing, 599
- active rotation matrix, 224
- actuation
 - system, 365
- actuator, 56
- adaptive control, 366
 - NDI-based, 200
- adaptive resampling, 570
- addition law of probability, 404
- adjoint
 - classical matrix, 26
- adjoint equation, 135
- adjoint vector, 134
- adjudate, 26
- aerodynamic chord
 - airfoil, 754
- aerodynamic parameter vector, 716
- aerodynamic twisting, 757
- aerostatic equation, 304
- affine transformation, 415
- aileron, 212
- aileron-rudder interconnect, 375
- air data, 366, 603
 - triplet, 366, 603
- air data system, 366
- air data systems, 603
- air density, 755
- airfoil, 754
- airplane
 - anatomy, 209
- airplane trim, 246
- airspeed, 235, 754
- alert limit, 611
- algebraic Riccati equation
 - continuous-time, 137
 - discrete-time, 185
- algebraic Riccati inequality, 147
- all-neighbors association, 684
- Allan deviation, 618
- Allan variance, 617
- alternate conditional sampling, 485
- alternate hypothesis, 441
- altimeter
 - barometric, 604
 - lidar, 602
 - pressure, 604
 - radar, 602
 - sensitive, 604
- altimeters, 604
- altitude correction
 - gravity, 298
- altitude hold, 367
- ambiguity fixing, 652
 - brute force, 655
 - geometry-free, 656
 - LAMBDA, 655
 - rounding, 655
- amplitude
 - doublet, 32
 - step, 31
- analog system, 12
- analysis plot
 - flight envelope, 261
 - v-n, 295
- angle
 - sweep, 757
- angle of attack, 237
 - airfoil, 754
- angular acceleration
 - orbital, 249
- three-dimensional, 226
- three-dimensional tensor, 229
- two-dimensional, 226
- angular velocity
 - orbital, 249
- three-dimensional tensor, 228
- three-dimensional vector, 226
- two-dimensional, 225
- two-dimensional tensor, 225

- aperture, 602
Aperture radar, 602
argument of periapsis, 247
argument of perigee, 247
aspect ratio
 wing, 756
association map, 733
associative laws
 set, 739
assumed density filters, 532
asymptotic slope, 44
asymptotically stable, 20
atmospheric delay
 code phase, 657
atmosphere, 660
atmospheric pressure, 603
atomic clocks, 615
attitude and heading reference system, 629
 air data, 629
attitude dynamics
 linearized gravity-gradient, 286
augmented proportional navigation guidance, 398
augmented rendezvous guidance, 398
augmented sigma-points
 prediction, 556
augmented signma-points
 correction, 557, 558
augmented state, 126, 494
augmented state covariance
 FI-SPKS/SPRTSS process, 558
 SPKF process, 556, 557
augmented state estimate
 FI-SPKS/SPRTSS process, 558
 SPKF process, 556, 557
auto-correlation
 PRN, 595
auto-correlation function, 428
 residual, 465
auto-covariance function, 428
auto-regressive process, 433
 vector, 433
auto-throttle, 377
autonomous multi-model, 526
autopilot, 209
availability
 information system, 611
available bandwidth, 91
avionics, 210
axiomatic probability theory, 403
axis
 lateral, 234
 local-horizontal, 234
 local-vertical, 234
 longitudinal, 234
axis-angle, 226
axisymmetric, 278
aymptotic stability, 128
azimuth, 638
backward-simulation particle smoother, 576
backwards recursive Bayes estimator, 492
ballistic coefficient, 712
ballistic flight
 boost phase, 710
 coast phase, 710
 reentry phase, 710
ballistic flight motion model, 710
bang-bang property, 141
bang-off-bang property, 143
bank angle, 236
barometric altimeter, 604
base
 receiver, 647, 652
base state, 435
baseline, 636, 647
baseline vector, 628
basic rotation matrices, 227
battery monitoring system, 606
Bayes estimator, 473
 parameter, 473
 state, 489
Bayes filter, 491
Bayes parameter estimator, 473
Bayes predictor, 490
Bayes risk, 473
Bayes smoother

- fixed-interval, 492
- Bayes state estimator, 489
 - median, 489
- Bayes' law
 - two events, 405
- Bayes' rule, 472
 - continuous random variables, 414
 - discrete random variables, 413
 - two events, 405
- Bayes' theorem
 - two events, 405
- Bayesian least-squares parameter estimator, 476
- Bayesian linear regression problem, 476
- Bayesian regression problem, 476
- beacon, 634
- beam-forming, 602
- bearing, 634
- beat frequency, 595, 600
- belief mass function
 - RFS, 448
- Bellman equation, 182
- Bernoulli distribution, 420
- Bernoulli random finite set, 450
- best linear unbiased estimator, 457
- between-epochs difference, 656
- bi-static radar, 599
- bias vector, 609
- binomial coefficient, 727
- biplane, 210
- birth
 - target, 678
- block diagram, 7
- blocked pitot tube, 604
- blocked static port, 605
- Bode gain-phase formula, 71
- Bode plot, 42
 - asymptotic slope, 44
 - DC gain, 43
- body-fixed frame, 234
 - mean-axis, 338
 - principal, 255
- boost phase, 710
- bootstrap particle filter, 567
- Borel set, 404
- boundary condition, 13
- bounded real lemma, 148
 - generalized, 165
- bounded set, 130
- Brownian motion, 432
- burn, 221
- cambered wing, 756
- camera
 - hyperspectral, 602
 - infrared, 602
 - mono-vision, 602
 - multippectral, 602
 - near-infrared, 602
 - stereo-vision, 602
 - visual, 602
- cameras, 602
- canard, 211
- candidate set, 441
 - measurement, 682
- cardinality, 446, 737
 - classification, 427
- cardinality distribution, 447
- cardinalized probability hypothesis density filter, 728
 - Gaussian mixture, 730
- carrier phase measurement
 - GPS, 595
- carrier phase positioning, 657
- carrier phase range equation
 - GNSS, 657
- cascade control, 364
- cascade interconnection, 54
- Cauchy problem, 126
- Cauchy's argument principle, 752
- Cauchy-Peano Theorem, 126
- Cayley-Hamilton definition, 114
- center of pressure
 - airfoil, 754
- central divided difference, 560
- central limit theorem, 423
 - generalized, 423

- central moment
random variable, 410
- Chapman-Kolmogorov equation
Bayesian state prediction, 491
Markov chain, 431
Markov jump process, 431
- characteristic equation
ODE, 25
SISO feedback control system, 63
state-space, 26
- characteristic exponent, 422
- characteristic function, 408
random vectors, 413
- characteristic polynomial
ODE, 25
SISO feedback control system, 63
state-space, 26
- Cholesky decomposition, 745
- chronometers, 615
- civil time, 617
- classic Euler angles, 227
- classical adjoint of matrix, 26
- classical control theory, 60
- classical optimal control problem, 134, 138
- clock
radio, 616
- clock drift, 617
receiver, 664
satellite, 659
- clock drift rate
satellite, 659
- clock error
satellite, 657
- clock network, 618
- clock offset, 618
satellite, 659
- clock skew, 618
- clocks, 615
atomic, 615
quartz, 615
- Clohessy-Wiltshire equations, 319
- closed set, 130
- closed-loop observer, 112, 177
- closed-loop transfer function, 61
- clutter, 678
- clutter rate
Poisson, 689
- clutter spatial density, 689
- co-domain
function, 737
- coast, 246
spacecraft, 221
- coast phase, 710
- cockpit
airplane, 210
helicopter, 216
- code division multiple access, 594
- code phase measurement
GPS, 595
- code phase positioning, 657
- coefficient of determination, 463
- colored-measurement-noise Kalman filter, 517
- column dimension, 740
- column rank, 740
- combination
ionosphere-free, 660
- commutative laws
set, 738
- compact set, 130
- complement
set, 737
- complementary cumulative distribution function, 408
- complementary filter, 495
- complementary sensitivity
transfer function, 62
- complex matrix, 740
- complex numbers, 737
- complex parametric uncertainty, 117
- complex vector, 739
- component build-up model, 268, 757
- computer vision, 602, 635
- conditional PDF, 414
- conditional PMF, 413
- conditional probability, 404
- cone, 145

- confidence ellipse
 - Gaussian, 425
- confidence interval
 - $100(1-\alpha)$, 465
- confidence level, 441
- confidence region
 - multivariate Gaussian, 425
 - univariate Gaussian, 421
- coning maneuver, 393
- conjugate prior, 474
- consistency
 - estimator, 454
- constant-mass approximation, 251
- constellation
 - satellite, 588
- constrained eigenvectors, 329
- continuously operating reference stations, 658
- control
 - automatic, 209
 - closed-loop, 57
 - feedback, 57
 - feedforward, 56
 - manual, 209
 - open-loop, 56
 - semi-automatic, 209
 - system, 365
- control and perception analysis, 10
- control and perception design, 10
- control and perception synthesis, 10
- control horizon, 191
- control input
 - nonlinear, 199
 - virtual LTI, 196
- control law, 10
- control moment gyro, 361
- control power
 - aileron, 776
 - elevator, 769
 - rudder, 776
- control theory
 - classical, 60
- control-affine dynamical system, 195
- controllability
 - mode, 29
 - output, 116
 - state, 114
 - under constraints, 116
- controllability Gramian
 - discrete-time, 179
 - LTI systems, 115
 - LTV systems, 131
- controllability grammian, 103
- controllability matrix, 115
- controllability test
 - Popov-Belevitch-Hautus, 115
- Controllable Canonical Form, 15
- controlled variable, 195
- conventional tail, 210
- convex set, 145
- convext set, 146
- convolution integral, 34
- convolutional neural network, 602
- Cooper-Harper Rating Scale, 273
- cooperating multi-model, 526
 - MAP base state estimator, 528
 - MAP mode sequence estimator, 528
 - MMSE base state estimator, 526
 - MMSE modal state estimator, 527
- coordinated flight, 260
- coordinated steady-flight condition, 245
- coordinated universal time, 617
- coordinated-turn motion model, 705
- copula, 419
- Coriolis effect, 586
- corner reflector, 599
- correction step, 679, 681
 - Bayes filter, 492
 - extended Kalman filter, 537, 540
 - information filter, 505
 - innovation-saturated extended Kalman filter, 543
 - iterative extended Kalman filter, 542
 - Kalman filter, 498
 - second-order extended Kalman filter, 546
 - sigma-point Kalman filter, 557
 - statistically linearized (Kalman) filter, 551

- correlated-noise Kalman filter, 514
 - continuous-time, 524
- correlated-noise Kalman-Bucy filter, 524
- correlation
 - random process, 430
- correlation coefficient, 417
- correlation matrix, 416
- cost function, 181
- cost functional, 133
- cost matrix, 136
 - cross-, 136
 - endpoint, 136
 - input, 136
 - state, 136
 - terminal, 136
- cost-to-come function, 189
- cost-to-go
 - continuous-time, 135
- cost-to-go function, 181
- costate equation, 135
- costate vector, 134
- countable, 737
- countably infinite set, 737
- counting numbers, 737
- counting process, 432
- covariance
 - sample, 475
- covariance intersection, 514
 - algorithm, 515
- covariance intersection Kalman filter, 515
- covariance matrix, 417
- covariance stationarity, 429
- critical lift, 716
- cross-correlation
 - function, 428
- cross-covariance
 - function, 429
 - noise, 511
- cross-covariance function
 - continuous-time noise, 522
- cross-covariance matrix, 417, 418
- crossover frequency separation, 364
- cruciform tail, 212
- cumulative distribution function, 407
 - joint, 412
- cycle count
 - fractional, 653
- cycle counting, 653
- damped least-squares, 468
- data association
 - k -nearest-neighbors, 685
 - all-neighbors, 684
 - global all-neighbor, 686
 - global nearest-neighbor, 685
 - nearest-neighbor, 684
- data association problem, 679
- DC gain, 43
- De Morgan's laws, 739
- dead reckoning, 633, 662
- death
 - target, 678
- decade
 - frequency, 44
- decibel table, 43
- decision altitude, 382
- decision height, 382
- decision theory, 132, 441, 453, 488
- decomposition
 - eigenvalue, 26
- deconvolution, 34
- decouples, 27, 173
- deep GNSS/INS integration, 673
- definiteness
 - matrix, 742
- deformable-body model, 240
- degeneracy problem, 566
- delay
 - dry, 660
 - hydrostatic, 660
 - wet, 660
- delay atmospheric
 - code phase, 657
- delay ionospheric
 - code phase, 657
- delay lock loop

- vector, 673
- delay margin, 66
- delay tropospheric
 - code phase, 657
- delayed gratification, 469
- derivative gain, 85
- descent direction, 454
- design matrix, 458
- detectability, 117
- detection
 - fault probability, 610
- detection theory, 442
- detector, 442
 - maneuver, 704
- deviation of the normal, 311
- diagonal matrix, 740
- diagonalizable, 26, 743
- difference
 - set, 738
- difference equations, 170
- differential game theory, 132
- differential pseudorange, 635
- differential range, 634
- differentiator
 - pure, 44
- diffuse prior, 689
- digital system, 12
- dilution of precision, 645
- dimension
 - row, 740
 - vector, 739
 - vector space, 740
- Dirac delta, 33
- direct method
 - Lyapunov, 129
- direction cosine matrix
 - three-dimensional, 226
 - two-dimensional, 223
- directional plane, 758
- discrete Fourier transform, 171
- discrete-time dynamics equation, 168
- discrete-time Fourier transform, 171
- discrete-time Kalman filter
 - one-step posterior, 500
 - one-step prior, 499
- discrete-time output equation, 168
- discrete-time transfer function, 170
- discretization, 180
- discretized LTI stochastic state-space model, 520
- discretized measurement noise covariance, 520
- discretized process noise covariance, 518
- disjoint set, 738
- disk margin, 67
- distance
 - Mahalanobis, 418
- distance measuring equipment, 383
- distances
 - statistical, 418
- distinct label indicator, 447
- distributive laws
 - set, 739
- divergence
 - Kullback-Leibler, 418
 - statistical, 418
- domain
 - function, 737
- Doppler effect, 635
- Doppler positioning, 664
- Doppler positioning equation, 664
- Doppler shift
 - GPS, 594
- double-differenced
 - cycle count range, 653
 - phase-based range, 652
 - pseudorange, 649
- doublet amplitude, 32
- doublet length, 32
- downwash angle, 758
 - equation, 762
- drag coefficient
 - wing, 755
 - wing, induced, 756
 - wing, parasitic, 756
 - wing, profile, 756
 - wing, wave, 756
 - wing, zero-lift, 756

- drag force
 - airfoil, 754
 - fixed-wing aircraft, 242
- drag parameter, 712
- dry delay, 660
- Dryden gust model, 306
- dual estimation, 494
- dual-spin stabilization, 379
- Dubins path OCP
 - 2D, 144
- Dubins path problem, 143
- Dubins vehicle, 143
- duplexer, 601
- dutch-roll mode, 270
- Dvoretzky-Kiefer-Wolfowitz inequality
 - multivariate, 413
 - univariate, 409
- dynamic inversion
 - LTI controller, 197
 - nonlinear controller, 199
- dynamic inversion control, 195
- dynamic pressure, 603
 - free-stream, 755
- dynamic programming, 182
- dynamic programming equation, 182
- dynamic uncertainty, 118
 - LTI, 118
- dynamic-controller feedback
 - discrete-time, 177
- dynamical system
 - control-affine, 195
 - linear-in-control, 195, 202
 - random, 434
 - stochastic, 434
- dynamics
 - equation, 12
- dynamics equation
 - discrete-time state-space, 168
 - state-space, 14
- Earth angular rate
 - WGS, 593
- Earth disk, 621
- Earth Gravitational Models, 298
- Earth gravitational parameter
 - WGS, 593
- Earth horizon sensor, 621
- Earth mean radius, 241
- Earth radius
 - mean, 318
- Earth rotation angle, 616
- Earth rotation rate, 669
- Earth sensor, 621
- Earth's radius of curvature
 - meridinal, 313
 - prime-vertical, 310
- Earth-centered, inertial, 231
- eccentricity, 247
 - WGS, 593
- eccentricity vector, 246
- effective navigation ratio, 397
- effective number threshold, 571
- effectiveness
 - aileron, 776
 - elevator, 769
 - rudder, 776
- efficient estimator, 457
- eigenfunctions, 322
- eigenvalue, 742
 - eigenvalue decomposition, 26, 742
 - eigenvalue equation
 - right, 26
- eigenvalue placement, 114
- eigenvalue problem, 742
- eigenvector
 - left, 742
- elastic stability and control derivative, 356
- elastic-body model, 240
- electric motor
 - geared, 359
- element
 - matrix, 740
 - set, 737
 - vector, 739
- elementary symmetric function, 729
- elevation, 638

- elevator, 210, 211
elevons, 211
ellipsoidal gates, 683
Ellipsoidal Gravity Formula
 WGS 84, 299
elliptical wing, 757
empennage, 210
empirical distribution function, 408, 613
 multivariate, 413
empty set, 737
endpoint cost, 134
engine
 airplane, 210
 rocket, 218
entry-wise matrix norm, 742
Epanechnikov kernel, 570
ephemeris, 596
ephemeris error
 satellite, 657
epoch, 247
equation of motion, 12
equations of motion
 Newton-Euler, 250
 Newton-Euler, rotating frame, 252
 rigid aircraft, 254
 rigid aircraft alternative, 255
 rotation equation, 250
 satellite attitude, 256
 six degrees-of-freedom, 252
 translation equation, 250
equator, 232
equatorial radius
 WGS, 592
equilibrium
 flight conditions
 rigid-body, 257
 point-mass flight conditions, 245
equilibrium point, 19, 169
 time-varying systems, 127
error
 estimator, 456
error dynamics
 LTI dynamic inversion, 196
nonlinear dynamic inversion, 199
error transfer function, 61
error-state correction step, 578
error-state extended Kalman filter, 578
error-state filter
 measurement update step, 579
 prediction step, 578
error-state Kalman filtering, 577
error-state measurement update step, 578
error-state time update step, 578
error-state vector, 577
estimator
 consistency, 454
 efficient, 457
 error, 456
 linear, 457
 maximum likelihood, 453
 mean, 456
 method of moments, 454
 variance, 456
estimator gain matrix, 457
Euclidean norm, 741
Euler angle
 attack, 237
 bank, 236
 flight-path, 236
 heading, 236
 inertial-to-body, 236
 navigation-to-body, 237
 navigation-to-wind, 236
 nutation, 236
 pitch, 237
 precession, 236
 roll, 237
 sideslip, 237
 spin, 236
 wind-to-body, 237
 yaw, 237
Euler angle ambiguity, 228
Euler angle rates
 $3 - 1 - 3$, 230
 $3 - 2 - 1$, 230
Euler angles, 227

- classic, 227
Tait-Bryan, 228
- Euler symmetric parameters, 226
- Euler's formula, 27
- event, 402
- event space, 403
- events
- elementary, 404
- exceedance, 408
- excess kurtosis, 421
- exit velocity, 299
- expectation
- random process, 428
- expectation function, 428
- expectation-maximization algorithm, 481
- expected average power
- random process, 430
- explained sum of squares, 464
- exponential function
- matrix, 25
 - univariate, 24
- extended Kalman Filter
- hybrid-time, 539
- extended Kalman filter
- continuous-time, 539
 - error-state, 578
 - iterative, 541
 - multiplicative, 579
 - second-order, 544
- extended Kalman particle filter, 571
- extended Kalman smoother
- fixed-interval, 547
- extended Kalman-Bucy filter, 539
- extended proportional navigation guidance, 400
- extended Rauch-Tung-Striebel smoother, 547
- extended target tracking, 679
- exteroceptive sensor, 440
- extreme value theory, 422
- F-test, 444
- fair, 403
- false alarm, 442
- probability, 611
- fast Fourier transform, 171
- fault, 609
- fault detection
- exclusion, 609
 - isolation, 609
- fault detection and isolation, 609
- fault vector, 609
- feasibility problem, 146
- feature, 634, 678
- feature tracker, 625
- feature tracking, 678
- feedback
- discrete-time dynamic-controller, 177
- feedback control system
- classical, 60
- feedback interconnection, 55
- feedback linearization control
- input-output, 195
- feedback linearization loop
- LTI, 196, 198
- feedthrough matrix, 14, 168
- filter
- SISO LTI, 58
- filter bank
- elementary, 525
- filtered derivative, 87
- filtering
- error-state, 629, 673
- final orbital errors, 659
- final value, 32
- final value theorem, 17
- finite burn, 222
- finite mixture distribution, 409
- finite set, 737
- finite-wing theory, 754
- fins, 218
- first-order approximation
- function, 20
- Fisher information matrix, 454
- fixed solution, 655
- fixed-gain controller, 176
- fixed-interval Fourier-Hermite Kalman smoother, 553

- fixed-interval Kalman filter, 547, 551
fixed-interval Kalman smoother, 500
fixed-interval quadrature Kalman smoother, 562
fixed-interval sigma-point Kalman smoothers, 556
fixed-interval state smoothing posterior PDF, 490
fixed-lag Kalman smoother, 507
fixed-lag state smoothing posterior PDF, 490
flap, 215
flat-Earth approximation, 242
flattening, 310
 WGS, 593
flicker frequency noise, 617
flicker noise, 600
flight condition
 engine-out, 259
flight dynamics and control, 209
flight envelope plot, 261
flight profile, 240
flight-path angle, 236
float solution, 655
flying qualities, 271
flywheel, 360
folded cumulative distribution function, 408
force
 aerodynamic, 242
 gravitational, 241
 propulsive, 241
 solar, 242
 thrust, 243, 769
forces
 fictitious, 226
forward value function, 189
Fourier inversion theorem, 41
Fourier series, 41
Fourier transform, 40
Fourier-Hermite Kalman filter, 553
Fourier-Hermite Rauch-Tung-Striebel smoother, 553
fractional cycles, 651
free air correction
 gravity, 298
free response, 31
LTI MIMO continuous-time, 99
free-stream airflow, 235
free-stream velocity
 airfoil, 754
 wind frame, 235
frequency
 angular, 41
 beat, 595
 intermediate, 595
 periodic, 41
frequency indexes, 465
frequency modulated continuous wave radar, 601
frequency modulated continuous-wave radar, 602
frequency modulation, 601
frequency resolution, 171
frequency response, 41
 gain, 40
 magnitude, 40
 phase, 40
frequency separation
 crossover, 364
frequentist probability theory, 402
Frobenius norm, 742
fuel monitoring system, 605
full rank, 740
function, 737
 co-domain, 737
 domain, 737
 invertible, 738
 mapping, 737
 one-to-one, 738
 range, 738
functional, 133
fundamental clock rate
 GPS, 594
fundamental transfer functions, 61
fuselage, 218
 airplane, 210
 helicopter, 216
fuselage frame, 258
gain
 frequency response, 40

- gain margin, 64
- gain scheduling, 131, 366
- Gamma function
 - multivariate, 426
- gate probability, 690
- gate volume, 683, 690
- gating step, 682
 - two-stage, 683
- gating threshold
 - ellipsoidal, 683, 690
 - rectangular, 683
- Gauss-Hermite cubature rule, 562
- Gauss-Hermite quadrature rule, 561
- Gauss-Markov process, 432
- Gauss-Markov theorem, 461
- Gauss-Newton algorithm, 467
- Gaussian distribution, 421
 - multivariate, 424
- Gaussian filtering, 532
- Gaussian process, 432
- Gaussian random walk, 518
- Gaussian sum filter, 580
- Gaussian-mixture
 - posterior approximation, 480
- geared electric motor, 359
- generalized calculus of variations, 134
- generalized control input, 109
 - discrete-time, 175
- generalized covariance intersection
 - algorithm, 515
- generalized disturbance, 109
 - discrete-time, 175
- generalized eigenvalue problem, 328
- generalized eigenvectors, 744
- generalized error, 109
 - discrete-time, 175
- generalized labeled multi-Bernoulli random finite set family, 451
- generalized least-squares
 - estimator, 466
- generalized least-squares problem, 461
- generalized LTI feedback controller, 109, 176
- generalized mass matrix, 331
- generalized plant, 109
- generating variate
 - scale mixture, 409
- geocentric orbits, 247
- geodesy, 232
- geodetic coordinates, 232
- geodetic frame, 234
- geographic position
 - celestial body, 634
- geoid, 232
- geometry matrix, 645
- geometry-free, 656
- geometry-free combination, 658
- geostationary orbit, 589
- geosynchronous orbit, 589
- Gibbs sampler, 485
- Gilvenko-Cantelli theorem, 409
- gimbal, 584
- gimbal lock, 585
- glide-slope, 383
- glide-slope guidance system, 385
- global all-neighbor data association, 686
- global hypothesis
 - MHT, 694
- global nearest-neighbor association, 685
- global positive definite, 129
- global stability, 128
- global uniform asymptotic stability, 128
- globally asymptotically stable, 20
- globally negative definite, 129
- globally negative semi-definite, 129
- globally positive semi-definite, 129
- globally stable, 20
- GNSS
 - carrier phase range equation, 657
 - pseudorange equation, 657
- GNSS/INS integration
 - deep, 673
 - loose, 672
 - tight, 672
- Gramian
 - controllability, 115
 - observability, 117

- gravitational harmonic second-order, 711
- gravitational law Newton's, 590
- gravity Somigliana, 670
- gravity model ellipsoidal-Earth, 711 flat-Earth, 711 spherical-Earth, 241, 711
- gravity turn motion model, 717 rocket, 717
- gravity vector, 298
- gravity-gradient moment linearized, 286
- gravity-gradient stabilization regions, 288
- great circle, 367
- Greenwich mean time, 634
- ground range, 638
- ground segment, 589
- ground speed, 236
- guidance system, 365
- guidance law augmented proportional navigation, 398 augmented rendezvous, 398 extended proportional navigation, 400 line-following, 366 proportional navigation, 397 rendezvous, 397
- guidance, navigation, and control, 364
- gust, 306
- gust model Dryden, 306 von Kármán, 306
- gyrocompassing, 669
- gyroscope, 584, 623 gyrostabilized, 584, 623, 624 rate, 624 strapdown, 584, 623, 624
- gyroscopic precession, 584
- gyrostabilized gyroscope, 584, 623, 624
- gyrostat, 379
- gyrotorquer, 361
- H-tail, 212
- Hamilton-Jacobi-Bellman equation, 135
- Hamiltonian, 134
- handling qualities, 271
- hard-iron variation, 623
- hardware-defined radio, 594
- hardware-in-the-loop, 10
- hazardously misleading information, 610 probability, 610
- heading angle, 236
- heading hold, 368
- heavy-tailed, 563
- height above average terrain, 603
- height above ground level, 603
- helicopter anatomy, 215
- Hermitian matrix, 740
- Hermitian transpose, 740
- hidden Markov model assumption, 489
- hidden Markov models, 435
- high Earth orbit, 589
- higher-order terms, 20
- higher-order unscented transform, 559
- Hill equations, 319
- Hill's frame, 318
- hold control altitude, 367 heading, 368 speed, 367
- holding time, 432
- homogeneous solution ODE, 24
- horizontal tail efficiency, 763 volume ratio, 763
- horology, 615
- Hungarian algorithm, 685
- Hurwitz matrix, 743
- hybrid process, 435
- hybrid-state estimation, 524

- hybrid-time feedback control system, 180
hydrostatic delay, 660
hyper-parameters
 conjugate prior, 474
hyperspectral camera, 602
hypotheses, 441
hypothesis
 alternate, 441
 null, 441
hypothesis test
 χ^2 -test, 443
 t-test, 442
 F-test, 444
 Z-test, 442
hypothesis testing problem, 441
hypothesis-oriented multi-hypothesis tracking, 695
- identity matrix, 15
IID cluster random finite set, 449
illumination, 599
image, 599
image processing, 635
images, 635
importance function, 564
importance sampling Monte Carlo, 486
importance weights
 normalized, 564
impulse function, 33
impulsive maneuver, 221
IMU grade
 automotive, 587
 consumer, 587
 industrial, 587
 marine, 587
 navigation, 587
 tactical, 587
inclination, 247
inclusion-exclusion principle, 404
increment, 427
incremental nonlinear dynamic inversion
 controller, 199
indefinite matrix, 743
- independence
 events, 405
 pairwise, 417
 random process, 429
independent and identically distributed, 454
independent increments, 429
index set, 427
indicator, 408
indirect method
 Lyapunov, 128
induced matrix norm, 742
induced norm
 system, 102
inertia
 matrix, 250
 moments, 250
 products, 250
 tensor, 250
inertia matrix
 aircraft, 254
 stability frame, 259
inertia tensor
 satellite, 255
inertia wheel, 360
inertial measurement units, 586
inertial sensors, 583
infinite dimensional problem, 322
information filter, 504
information gain, 418
information matrix, 504
information set, 687
information state, 687
infrared camera, 602
inherent errors
 ADS, 605
initial condition, 13
initial condition response, 31
initial value problem, 31, 126
initialization step
 BPF, 567
 EKF, 572
 extended Kalman filter, 536, 539

- innovation-saturated extended Kalman filter, 543
- iterative extended Kalman filter, 541
- Kalman filter, 498
- RBPF, 574
- second-order extended Kalman filter, 546
- sigma-point Kalman filter, 556
- SPPF, 572
 - statistically linearized (Kalman) filter, 550
- inner product, 727
- inner-outer loop control, 364
- innovation covariance, 498
- innovation process, 498
- innovation-saturation extended Kalman filter, 542
- input
 - doublet, 32
 - impulse, 33
 - signal, 13
 - step, 31
 - vector, 14, 168
- input matrix
 - continuous-time, 14
 - discrete-time, 168
- input-output feedback linearization control, 195
- instability
 - LTI system, 28, 173
 - modal continuous-time, 28
 - modal discrete-time, 173
- instant cost, 134
- instrument guidance system, 383
- instrument landing systems, 383
- integer ambiguity, 651
- integral quadratic constraint, 118
- integration-points, 483
- integrator
 - pure, 46
- integrity monitor, 610
- integrity monitoring system, 610
- intensity, 720
- intensity function, 449
- interacting multi-model
 - filter, 529
 - Kalman filter, 529
- intercept guidance problem, 394
- intercept problem, 396
- interior point method, 146
- intermediate frequency, 595
- internal model principle, 123
- intersection
 - set, 738
- inverse
 - matrix, 740
- inverse distribution function, 408
 - generalized, 408
- inverse transform sampling, 419
- inverted V-tail, 213
- invertible
 - function, 738
- ionosphere-free combination, 660
- ionospheric delay
 - code phase, 657
- ISMC estimator, 486
- iterative extended Kalman filter, 541
- iterative least-squares
 - estimator, 467
- Jacobian, 21
- jerk model
 - white noise, 700
- jet-damping effect, 303
- joint CDF
 - random process, 428
- joint cumulative distribution function, 412
- joint estimation, 494
- joint PDF
 - random process, 428
- joint PMF
 - random sequence, 428
- joint probabilistic data association filter, 691
 - coupled, 692
- joint probability density function, 412
- joint probability mass function, 412
- joint state, 494
- Jordan blocks, 743
- Jordan canonical form, 27, 744
 - discrete-time, 173

- Jordan Chains, 100
- Joseph form, 499
- Julian date, 617
- Julian day number, 617
- jump-linear system, 435
- Kalman filter
 - colored-measurement-noise, 517
 - continuous-time, 522
 - correlated-noise, 514
 - covariance intersection, 515
 - nonlinear, 532
 - quadrature, 562
 - sigma-point, 556
 - square-root, 505
- Kalman gain
 - continuous-time, 436
 - discrete-time, 437
 - hybrid-time, 438
- Kalman smoother
 - fixed-interval, 500
 - fixed-interval Fourier-Hermite, 553
 - fixed-interval quadrature, 562
 - fixed-interval sigma-point, 556
 - fixed-lag, 507
- Kalman-Bucy filter, 522
- Kalman-Yacubovich-Popov lemma, 148
- Kepler's equation, 597
- Keplerian elements, 247, 597
- kernel
 - Epanechnikov, 570
- kernel PDF, 570
- keying
 - amplitude-shift, 593
 - frequency-shift, 593
 - phase-shift, 593
- kinematic, 647
- Kolmogorov-Smirnov statistic, 408
- Krasovskii-LaSalle theorem, 130
- Kullback-Leibler divergence, 418
- kurtosis, 411
 - excess, 421
- L-band, 593
- L-moment, 455
- labeled IID cluster random finite set, 451
- labeled multi-Bernoulli random finite set, 451
- labeled Poisson random finite set, 451
- labeled random finite set, 447
 - δ -generalized multi-Bernoulli, 733
 - generalized multi-Bernoulli family, 451
 - IID cluster, 451
 - multi-Bernoulli, 451
 - Poisson, 451
- Lagrangian term, 134
- LAMBDA method
 - baseline-constrained, 628
- Laplace transform, 16
 - inverse, 16
- latent variables
 - Rao-Blackwellized particle filter, 573
- lateral plane, 759
- latitude correction
 - gravity, 299
- latitude, longitude, altitude, 232
- law of gravitation, 241
- law of total probability
 - continuous random variables, 414
 - events, 405
- lead angle, 395
- leap second, 617
- learning rate, 454
- least upper bound, 742
- least-squares
 - constrained regression problem, 469
 - damped, 468
 - iterative, 467
- least-squares estimator, 461
 - nonlinear, 467
- least-squares problem
 - generalized, 461
 - ordinary, 461
 - weighted, 461
- least-squares regression problem, 461
 - linear, 461
- left half plane, 28, 173
- level, 291

- level flight, 246
level set, 130
leveling, 669
Levenburg-Marquardt algorithm, 468
 modified, 468
lever arm
 GNSS/INS, 677
LFT inverse
 lower, 108
 upper, 108
lidar, 602
lidar altimeter, 602
Lie derivative, 198
lift coefficient
 wing, 755
lift force
 airfoil, 754
 fixed-wing aircraft, 242
lift-curve slope
 airfoil, 757
 wing, 756
lifting-line theory, 757
likelihood, 405
likelihood function
 multi-target, 721
likelihood-ratio, 444
 simple-vs.-simple, 445
likelihood-ratio test
 generalized, 444
 simple-vs.-simple, 445
limiter, 369
line-of-sight, 638, 645
line-of-sight angle, 395
linear fractional transformation, 104
 lower, 104
 upper, 105
linear least-squares estimator, 461
linear least-squares regression problem, 461
linear matrix equality, 746
linear matrix inequality, 746
 strict, 746
linear objective problem, 146
linear parameter estimator, 457
linear time-invariant, 13
linear velocity
 apparent, two-dimensional, 225
 inertial, two-dimensional, 224
linear, parameter-varying system, 131, 164
linear-in-control dynamical system, 195, 202
linear-quadratic estimator, 150
 continuous-time, 436, 522
 discrete-time, 437, 438
linear-quadratic regulator, 136
 extended, 190
 iterative, 188
linear-quadratic-Gaussian OCP, 150
linearization
 cosine, 23
 error, 20
 Jacobian, 21
 sine, 23
 theorem, 20
 univariate, 20
linearized dynamics
 gravity-gradient attitude, 286
linearized LTI state-space model
 discrete-time, 170
linearized rigid airplane dynamics
 lateral-directional, 267
 longitudinal, 267
lines-of-position, 634
Lipschitz condition, 127
lithographic construction, 586
local tangent plane (LTP), 232
local-vertical, local-horizontal frame
 nadir-pointing, 234
localization, 633
localizer, 383
localizer guidance system, 387
localizer-type direction aid, 383
locally negative definite, 129
locally negative semi-definite, 129
locally positive definite, 129
locally positive semi-definite, 129
location, 633
log-likelihood ratio

- target tracking, 697
- long-period mode, 269
- longitude of ascending node, 247
- longitudinal plane, 757
- loop bandwidth, 71
- loop phase error, 619
- loop-shaping
 - control stages, 70, 76
 - mixed-sensitivity \mathcal{H}_∞ , 159
 - SISO, 70
- loop-shaping design, 77
- loop-shifting, 110
- loose GNSS/INS integration, 672
- low Earth orbit, 589
- lower triangular matrix, 740
- LQR-smoothing, 190
- LTI dynamic inversion controller, 197
- LTI MIMO continuous-time stability, 100
- LTI MIMO free response
 - continuous-time, 99
- LTI stochastic state-space model
 - discretized, 520
- LTI system
 - normality, 140
- Luenberger observer equation
 - continuous-time, 436
 - discrete-time, 437
 - hybrid-time, 438
- Luenberger observer matrix, 112, 178
- lumped-mass model, 322
- Lyapunov function, 129
- Lyapunov function candidate, 129
- Lyapunov stability of equilibrium point
 - time-varying system, 128
- Lyapunov's direct method, 129
- Lyapunov's indirect method, 128
- machine vision, 602
- magnetic compass, 622
- magnetometer, 622
 - three-axis, 623
- magnitude
 - frequency response, 40
- Mahalanobis distance, 684
- maneuver
 - time constant, 701
- map
 - localization, 633
- mapping
 - function, 737
- marginal likelihood, 472
- marginal parameter posterior PDF, 493
- marginal PDF, 414
- marginal stability
 - LTI system, 28, 173
 - modal continuous-time, 28
 - modal discrete-time, 173
- marginal state posterior PDF, 493
- marginalized particle filter, 574
- marker beacons, 383
- Markov chain, 430
- Markov chain Monte Carlo, 485
- Markov jump system, 435
- Markov jump-linear system, 435
- Markov jump-mean acceleration model, 703
- Markov process, 430
- Markov property, 430
 - n^{th} -order, 433
- Marquardt parameter, 468
- mask angle, 660
- mass matrix
 - generalized, 331
- master control station, 590
- matched uncertainties, 196
- mathematical optimization, 132
- mathematical programming, 132
- matrix
 - correlation, 416
 - covariance, 417
 - cross-covariance, 417, 418
 - diagonal, 740
 - element, 740
 - Hermitian, 740
 - lower triangular, 740
 - norm, 741
 - notation, 740

- orthogonal, 740
- square, 740
- symmetric, 740
- unitary, 740
- upper triangular, 740
- variance-covariance, 417
- matrix decomposition
 - Cholesky, 745
 - eigenvalue, 742
 - QR, 745
 - singular value, 744
- matrix exponential
 - Cayley-Hamilton, 114
- matrix exponential function, 25
- matrix inverse, 740
- matrix inversion lemma, 477, 740
- matrix Lyapunov equation, 104
- matrix rank, 740
- matrix transpose, 740
 - conjugate, 740
 - Hermitian, 740
- maximum *a posteriori* estimator, 473
- maximum likelihood estimator, 453
- maximum modulus theorem, 96
- maximum norm
 - matrix, 742
 - vector, 741
- maximum overshoot, 32
- maximum posterior
 - Bayes state estimator, 489
- maximum singular value, 745
- Mayer term, 134
- mean, 411
 - estimator, 456
 - function, 428
 - random process, 428
 - random vector, 416
 - sample, 474
- mean radius of Earth, 318
- mean-axis constraints, 338
- measure, 404
- measurement
 - candidate set, 682
 - measurement equation, 434
 - multi-target, 721
 - measurement error, 458
 - measurement noise, 434
 - measurement noise covariance
 - discretized, 520
 - measurement origin uncertainty, 678, 679
 - measurement update step, 679, 681
 - error-state, 578
 - median, 411
 - Bayes estimator, 473
 - Bayes state estimator, 489
 - medium Earth orbit, 589
 - member
 - set, 737
 - memoryless process, 430
 - method of moments estimator, 454
 - metric, 418
 - Metropolis-Hastings algorithm, 569
 - MHA-MCMC parameter estimator, 486
 - micro-electro-mechanical systems, 586
 - micro-electromechanical systems, 586
 - minimal realization, 17
 - minimum mean square error, 456
 - Bayes state estimator, 489
 - minimum mean-square error
 - Bayes parameter estimator, 473
 - minimum phase system, 71
 - minimum singular value, 745
 - minimum variance unbiased estimator, 456
 - minimum-energy optimal control problem, 139
 - minimum-fuel optimal control problem, 142
 - minimum-time optimal control problem, 139
 - mis-identified fault
 - probability, 610
 - missed detection, 442
 - hazardously misleading information, 610
 - missions, 271
 - mixed-sensitivity \mathcal{H}_∞ loop-shaping, 159
 - mixture
 - PDFs, 688
 - modal analysis, 100
 - modal coordinates, 322

- modal state, 435
- mode, 100, 411
 - system, 26
- mode approximation
 - phugoid, 270
 - roll, 270
 - short-period, 269
 - spiral, 271
- mode shapes, 322
- mode-finding algorithm, 480
- model
 - deformable-body, 240
 - elastic-body, 240
 - point-mass, 240
 - rigid-body, 240
- model evidence, 472
- model predictive control, 133, 191
- model set, 525
- model-based design, 7
- modulation
 - binary code, 593
 - binary offset carrier, 593
 - phase, 593
 - split spectrum, 593
- moment
 - aerodynamic, 242
 - propulsive, 241
 - random variable, 410
 - solar, 242
- moment-generating function, 407
 - random vectors, 413
- moments of inertia
 - principal, 256
- momentum exchange devices, 360
- momentum wheel, 360
- monitor stations, 590
- monitoring system
 - battery, 606
 - fuel, 605
 - integrity, 610
- mono-static radar, 599
- mono-vision camera, 602
- Monte Carlo
 - importance sampling, 486
 - Markov chain, 485
 - methods, 483
 - rejection sampling, 484
- Moore-Penrose inverse, 462
- most powerful detector, 446
- motion model
 - ballistic flight, 710
 - coordinated-turn, 705
 - general polynomial, 699
 - turning, 704
- multi-Bernoulli distribution, 421
- multi-Bernoulli mixture random finite set, 451
- multi-Bernoulli random finite set, 451
- multi-hypothesis tracking, 694
 - hypothesis-oriented, 695
 - track-oriented, 695
- multi-modal, 563
- multi-model
 - autonomous, 526
 - cooperating, 526
 - target tracking, 704
 - variable-structure, 526
- multi-model filtering, 525
 - interacting, 529
- multi-sensor information fusion, 494
- multi-sensor system, 494
- multi-static radar, 599
- multi-target filtering density, 721
- multi-target measurement equation, 721
- multi-target posterior density, 721
- multi-target process equation, 720
- multi-target transition density, 721
- multi-tracker fusion, 679
- multiangulation, 634
- multiangulation, 634
- multilateration, 634, 643
- multiple comparisons problem, 441
- multiple testing problem, 441
- multiplicative extended Kalman filter, 579
- multispectral camera, 602
- Murty's algorithm, 686
- mutual orthogonality

- random process, 430
- mutually exclusive, 404, 738
- mutually independence
 - random processes, 430
- mutually independent, 406

- nadir, 620
- National Marine Electronics Association, 659
- natural numbers, 737
- Navigation, 633
- navigation, 365, 678
 - system, 365
- navigation frame, 233
- navigation state, 633, 667, 678
- navigation systems, 662
- navigational stars, 620, 627
- near-infrared camera, 602
- nearest-neighbor association, 684
- negative definite matrix, 743
- negative semi-definite matrix, 743
- nested-loop control, 364
- net acceleration, 712
- network
 - clock, 618
- Newton's second law, 241
- Newton-Raphson algorithm, 479
- Neyman-Pearson lemma, 446
- no-wind approximation, 254
- noise, 488
 - flicker, 600
 - measurement, 434
 - process, 434
 - shot, 600
- noise cross-covariance, 511
- noise cross-covariance function
 - continuous-time, 522
- nominal-state vector, 577
- non-informative prior, 473
- non-isolable
 - fault probability, 611
- nonlinear dynamic inversion controller, 199
- nonlinear Kalman filtering, 532
- nonlinear least-squares estimator, 467

- nonlinear least-squares regression problem , 467
- norm
 - L_2 , 741
 - L_∞ , 741
 - L_p , 741
 - $L_{2,2}$, 742
 - $L_{\infty,\infty}$, 742
 - p , 741
- entry-wise matrix, 742
- Euclidean, 741
- Frobenius, 742
- induced matrix, 742
- matrix, 741
- matrix maximum, 742
- taxicab, 741
- vector, 740
 - vector maximum, 741
- normal distribution, 421
- normal load factor, 292
- normal LTI system, 140
- normal modes, 330
- normalized Butterworth polynomial, 80
- normalized importance weights, 564
- nose
 - airplane, 210
 - helicopter, 216
 - launch vehicle, 218
- null hypothesis, 441
- null set, 737
- numerical integration, 483
- nutation angle, 236
- Nyquist frequency, 171
- Nyquist plot, 41, 748
- Nyquist stability criterion
 - simplified SISO, 752
 - SISO, 752

- Oberth maneuver, 222
- object tracking, 678
- objective function, 181
- objective functional, 133
- oblate body, 280
- oblate spheroid, 232

- observability, 116
 - mode, 29
 - state, discrete-time, 179
- observability Gramian, 117
 - LTV systems, 131
- observability grammian, 104
- observability matrix, 116, 180
- observability test
 - Popov-Belevitch-Hautus, 116
- observation equation, 434
- observation error, 458
- observation matrix, 458
- observer feedback control, 112, 177
- observer feedback control system, 112, 178
- observer law, 10
- OCP
 - linear-quadratic, 136
- ODE solution
 - homogeneous, 24
 - particular, 24
- odometer, 662
- one-to-one
 - function, 738
- onset time, 704
- onto
 - function, 738
- open circuit voltage, 607
- open-loop observer, 112, 177
- open-loop transfer function, 60
- operator, 209
- optical gyroscopes, 585
- optimal control
 - function, 133
 - problem, 132
 - sequence, 181
- optimal control problem
 - fundamental stochastic, 436
 - linear-quadratic-Gaussian, 436
 - minimum-energy, 139
 - minimum-fuel, 142
 - minimum-time, 139
- optimal control theory, 132
- optimal estimation
 - parameter, 242, 265
 - optimal estimator
 - parameter, 453
 - optimal guidance law
 - intercept, 399
 - optimal information fusion
 - general, 494
 - linear-Gaussian, 494
 - optimal tradeoff curve, 153
 - optimality equation, 182
 - orbit
 - geostationary orbit, 589
 - geosynchronous, 589
 - high Earth, 589
 - low Earth, 589
 - medium Earth, 589
 - orbit injection, 221
 - orbit insertion, 221
 - orbit phasing manuever, 222
 - orbital angular momentum, 246
 - orbital elements, 247
 - orbital mean motion, 249
 - orbital mechanics, 246
 - order statistic, 455
 - ordinary differential equation, 13
 - linear, 14
 - linear time-invariant, 15
 - order, 13
 - proper, 13
 - time-invariant, 14
 - unforced, 13
 - ordinary least-squares
 - estimator, 462
 - ordinary least-squares problem, 461
 - orthogonal matrix, 740
 - orthogonality
 - random processes, 430
 - Oswald efficiency
 - wing, 756
 - Oswald's efficiency
 - effective, 243
 - outcome space, 403
 - outcomes

- frequentist, 402
- outer tracking loop
 - LTI, 196, 198, 199
- output
 - signal, 13
 - vector, 14, 168
- output controllability, 116
- output equation, 14, 168
- output feedback control, 111, 176
- output matrix, 14
 - discrete-time, 168
- overbound, 612
 - one-sided, 612
 - two-sided, 612
- overbounding, 612
 - error, 611
- parallel interconnection, 54
- parameter estimation, 453
 - problem, 453
- parameter estimator, 453
 - BLS, 478
 - ISMC, 486
 - least-squares, 476
 - MHA-MCMC, 486
 - optimal, 453
 - RSMC, 484
 - VB, 482
- parameter vector, 131, 164
- parametric uncertainty, 117
 - complex, 117
 - real, 117
- Parseval's theorem, 154
- partial-fraction decomposition, 17
- particle depletion, 568
- particle filter, 563
 - extended Kalman, 571
 - extended/sigma-point, 571
 - marginalized, 574
 - Rao-Blackwellized, 574
 - sigma-point, 571
- particle smoother
 - backward-simulation, 576
- reweighted, 576
- particular solution
 - ODE, 24
- partition
 - set, 738
- passive
 - energy sensing, 599
- passive rotation quaternion, 226
- PDAF
 - prediction step, 689
- peak time, 32
- perception, 439
- perception systems, 439
- perfect-gas equation, 304
- performance index, 133
- perifocal frame, 247
- period
 - function, 39
- permutation coefficient, 727
- perturbation
 - continuous-time vector, 21
 - discrete-time vector, 169
 - form, 20
 - scalar, 20
- phase
 - frequency response, 40
- phase ambiguity
 - receiver, 651
 - transmitter, 651
- phase delay
 - receiver, 651
 - transmitter, 651
- phase margin, 65
- phase wind-up, 651, 658
- phase-based range equation, 651
- phases of flight
 - aircraft, 219
 - spacecraft, 220
- phugoid mode, 269
- PI control
 - with rate feedback, 88
- PID
 - interacting form, 87

- parallel form, 85
- proportional gain, 85
- series form, 87
- standard form, 86
- PID control
 - integral gain, 85
- PIDF control
 - filtered series form, 87
 - filtered standard form, 87
 - parallel form, 87
- pilot, 209
- pitch angle, 237
- pitch damper, 270
- pitching moment
 - airfoil, 755
- pitching moment coefficient
 - wing, 755
- pitot pressure, 603
- pitot tube
 - blocked, 604
- plan, 366
- planning, 366
 - mission, 366
 - path, 366
 - system, 366
 - trajectory, 366
- planning, control, and perception, 8
 - aerospace, 8
- plant, 9
- plant modeling, 9
- point cloud, 602
- point clouds, 635
- point target tracking, 679
- point-mass model, 240
- Pointryagin's principle, 135
- Poisson distribution, 420
- Poisson process, 432
 - spatial, 689
- Poisson random finite set, 449
- polar form, 40
- polar radius
 - WGS, 593
- pole placement, 114
- pole-zero cancellation, 17
- pole-zero plot, 29
- poles
 - system, 26
- polynomial-matrix model, 19, 296
- Popov-Belevitch-Hautus controllability test, 115
- Popov-Belevitch-Hautus observability test, 116
- pose, 633
- position-domain integration, 672
- positioning
 - carrier phase, 657
 - code phase, 657
 - Doppler, 664
- positioning, navigation, and timing, 633
- positive definite matrix, 743
- positive semi-definite matrix, 743
- posix time, 617
- possibility space, 403
- posterior PDF, 472
 - state filtering PDF, 490
 - state prediction, 490
 - state smoothing, 490
- posterior probability, 405
- power
 - detector, 446
- power spectral density, 429
- PPP
 - random finite set, 449
- precession
 - prograde, 280
 - retrograde, 280
- precession angle, 236
- precision approach, 382
- precision matrix, 417
- prediction, 488
- prediction horizon, 191
- prediction step
 - Bayes filter, 491
 - extended Kalman filter, 536, 540
 - fixed-lag Kalman smoother, 510
 - information filter, 505
 - innovation-saturated extended Kalman filter, 543

- iterative extended Kalman filter, 541
Kalman filter, 498
PDAF, 689
second-order extended Kalman filter, 546
sigma-point Kalman filter, 556
square-root Kalman filter, 505
statistically linearized (Kalman) filter, 550
- pressure
atmospheric, 603
dynamic, 603
pitot, 603
ram, 603
stagnation, 603
static, 603
total, 603
- pressure altimeter, 604
- prime meridian, 232
- principal moments of inertia, 256
- principle of optimality, 135, 181
- principle of superposition, 12, 14
- prior editing, 568
- prior PDF, 472
- prior probability, 405
- probabilistic data association filter, 688
joint, 691
joint coupled, 692
- probability
survival, 720
- probability density function, 407
joint, 412
- probability distribution
 K , 424
 Γ , 423
(student's) t , 423
Bernoulli, 420
Cauchy, 422
central χ^2 , 423
double exponential, 424
exponential, 423
finite mixture, 409
Gaussian, 421
Gaussian scale mixture, 426
Gaussian-inverse-Wishart, 427
- Inverse-Wishart, 426
Laplace, 424
Levy, 422
Levy α -stable, 422
multi-Bernoulli, 421
noncentral χ^2 , 423
normal, 421
Pareto, 422
Poisson, 420
scale mixture, 409
scaled inverse- χ^2 , 423
standard Cauchy, 422, 423
symmetric α -stable, 422
uniform, 419
Wishart, 426
- probability generating functional (PGFL), 448
- probability hypothesis density, 449, 720
particle, 723
- probability hypothesis density filter, 723
cardinalized, 728
Gaussian mixture, 725
Gaussian mixture-cardinalized, 730
- probability mass function, 406
joint, 412
- probability of exceedance, 307
- probability space, 404
- probability theory
axiomatic, 403
frequentist, 402
- probability weighted moments, 455
- probability-generating function
discrete random variable, 406
random vectors, 413
- problem
hypothesis testing, 441
multiple comparisons, 441
multiple testing, 441
optimal control, 132
parameter estimation, 453
- process equation, 434
multi-target, 720
- process noise, 434
- process noise covariance

- discretized, 518
- prograde precession, 280
- prolate body, 280
- proof mass, 583, 621
- propeller, 210
- proper subset, 737
- proportional navigation guidance, 397
- proportional-integral-derivative control, 85
- proposal distribution, 419
- proposal functions, 484
- proposal PDF, 484
 - PF, 564
- proprioceptive sensor, 440
- protection level, 611
- pseudo-random number generation, 419
- pseudoazimuth, 678
- pseudoelevation, 678
- pseudoinverse, 180, 198, 462
 - left, 462
- pseudomeasurements, 678
- pseudometric
 - statistical, 418
- pseudorange, 635
 - differential, 635
 - measurement, 644
- pseudorange equation
 - base, 648
 - GNSS, 657
 - rover, 647
 - simplified, 644
 - un-differenced, 644
- pseudorange measurement
 - GPS, 595
- pseudorange rate, 663, 678
 - measurement, 664
- pseudorange rate equation, 664
- pseudoranges, 678
- pulse compression, 601
- pulse repetition frequency, 601
- pusuit-evasion guidance problem, 394
- QR decomposition, 745
- quadratic programming, 469
- quadrature, 483
- quadrature Kalman filter, 562
- quadrature Rauch-Tung-Striebel smoother, 562
- quantile function, 408
- quartz clocks, 615
- quasimetric
 - statistical, 418
- quaternion
 - passive rotation, 226
- radar, 634
 - bi-static, 599
 - duplexer, 601
 - frequency modulated continuous wave, 601
 - illumination, 599
 - mono-static, 599
 - multi-static, 599
 - ultra-wideband, 600
- radar altimeter, 602
- radar cross section, 599
- radar Doppler shift, 600
- radar range equation, 600
- radially unbounded, 130
- radio clock, 616
- radio technical commision for maritime services, 658
- radius of curvature
 - east-west, 668, 674
 - instantaneous, 245
 - meridian, 668
 - north-south, 668, 674
 - prime vertical, 668
- ram pressure, 603
- random dynamical system, 434
- random finite set, 446
 - Bernoulli, 450
 - birth, 720
 - clutter, 721
 - detection, 721
 - IID cluster, 449
 - intensity function, 449
 - labeled, 447
 - multi-Bernoulli, 451

- multi-Bernoulli mixture, 451
- Poisson, 449
- PPP, 449
- spawn, 720
- survival, 720
- random finite sets
 - BMF, 448
- random gusts, 306
- random motion model, 666
- random number generation, 419
- random process, 427
 - auto-regressive, 433
 - Brownian motion, 432
 - colored noise, 429
 - counting, 432
 - expected average power, 430
 - Poisson, 432
 - standard Wiener, 433
 - strictly white noise, 429
 - white noise, 429
 - white noise Gaussian process, 432
 - Wiener, 432
- random sequence, 427
- random variable, 406
 - central moment, 410
 - continuous, 406
 - discrete, 406
 - kurotsis, 411
 - mean, 411
 - median, 411
 - mixed, 409
 - mode, 411
 - moment, 410
 - skewness, 411
 - standard deviation, 411
 - standardized moment, 411
 - variance, 411
- random variate, 419
- random variate generation, 419
- random vector, 412
- random walk, 518
- range, 634
 - differential, 634
- function, 738
- ground, 638
- slant, 638
- range rate, 634
- range rate measurement
 - GPS, 595
- range-domain integration, 672
- rank
 - column, 740
 - deficient, 740
 - full, 740
 - matrix, 740
 - row, 740
- Rao-Blackwell theorem, 573
- Rao-Blackwellized particle filter, 574
- rapid orbital errors, 659
- rate feedback control, 88
- rational numbers, 737
- Rauch-Tung-Striebel smoother, 500
 - Fourier-Hermite, 553
 - quadrature, 562
 - sigma-point, 556
- reachability
 - state, 114
- reaction wheels, 360
- real matrix, 740
- real numbers, 737
- real parametric uncertainty, 117
- real vector, 739
- real-time networks, 658
- realization, 406
 - random process, 427
 - vector, 412
- receding horizon control, 191
- receding-horizon control, 133
- receiver clock drift, 664
- rectangular gates, 683
- recursive Bayes estimator, 491
- recursive least-squares, 476
- reduced-order model, 29
- reentry flight, 710
- reference ellipsoid, 232, 309
- reference frame

- Earth-centered, Earth-fixed, 231
east-north-up, 233
Hill's frame, 318
international celestial reference frame, 231
international terrestrial reference frame, 232
north-east-down, 233
rotations, 222
velocity-turn-climb, 715
reference frames, 222
 Earth-centered, 231
reference model, 200
reference trajectory, 240
reference transmitter, 646
region of attraction, 129
region of convergence, 170
regression error, 458
regression matrix, 458
regression model, 460
regression problem, 460
 Bayesian linear, 476
 nonlinear least-squares, 467
regression sum of squares, 464
regression weights, 553
regressor, 460
regulator, 136
 controller, 123
Reid's algorithm, 695
rejection sampling, 419
rejection sampling Monte Carlo, 484
relative entropy, 418
relative frequency
 probability, 402
remote sensing, 440
rendezvous and proximity operation, 318
rendezvous guidance, 397
rendezvous maneuver
 spacecraft, 222
rendezvous problem, 396
rendezvous, proximity operations, docking, and
 undocking, 222
requirements
 design, 10
resampling
adaptive, 570
low-variance, 567
multinomial, 566
regularized, 569
remainder, 566
residual, 566
stratified, 567
systematic, 567
residual, 460
residual sum of squares, 463
residue, 17
resolvent, 18
resonant
 peak, 52
resonant valley, 51
resonator, 615
restricted-step methods, 469
retrograde precession, 280
reweighted particle smoother, 576
RFS
 cardinality distribution, 447
 density, 447
 probability generating functional (PGFL),
 448
RFS moment density, 449
 N^{th} -order, 449
 first-order, 449
Riccati matrix
 continuous-time, 137
 discrete-time, 185
right ascension of ascending node, 247
right hand circular polarized, 593
rigid-body model, 240
rise time, 32
RMS gust intensities, 306
robust stability, 120
robust stability margin, 120
robustness
 SISO feedback control system, 64
roll angle, 237
roll damper, 271
roll mode, 270
roll-spiral mode, 270

- rolling moment
wing, 755
- rolling moment coefficient
wing, 755
- root locus, 84, 373
- root mean square, 618
- rotation angle
Euler axis, 226
- rotation matrix
passive, 224
- rotor mast, 216
- roughening, 568
- rover
receiver, 647, 652
- row rank, 740
- RSMC estimator, 484
- rudder, 210, 211
launch vehicle, 218
- ruddervator, 215
- running cost, 134
- Sagnac effect, 585
GNSS, 659
- sample, 427
- sample covariance, 475
- sample function, 427
- sample impoverishment, 568
- sample mean, 474
- sample path, 427
- sample point, 404
- sample space, 403
- sample trajectory, 427
- sampling
Markov chain Monte Carlo, 569
pseudo-random number, 419
random number, 419
- sampling frequency, 171
- sampling importance resampling, 566
- sampling interval, 171
- satellite constellation, 588
- satellites, 219
- saturation bound, 543
- saturation function, 542
- scalar
notation, 739
- scale mixture random variable, 409
- scale parameter
scale mixture, 409
- Schur Complement Lemma, 747
- score
likelihood, 454
- search vector, 467
- second-order extended Kalman filter, 544
- seismic mass, 583, 621
- semi-lactus rectum, 248
- semi-major axis, 247
- semi-Markov jump process, 703
- semidefinite programming, 148
- semidefinite programs, 145
- semimetric
statistical, 418
- sensitive altimeter, 604
- sensitivity
transfer function, 61
- sensitivity tradeoff, 90
- sensor, 57, 439
exteroceptive, 440
proprioceptive, 440
remote, 440
system, 365
- sensor data, 679
- sensor error
corrective, 440
predictive, 440
- sensor networks, 679
- separation principle, 113, 178
LQG, 436
stochastic control, 439
- sequential importance resampling, 566
- sequential importance sampling with resampling, 566
- sequential Kalman filter
discrete-time, 503
- serial interconnection, 54
- servomechanism state-space model, 125
- set

- associative laws, 739
commutative laws, 738
complement, 737
countably infinite, 737
De Morgan's laws, 739
disjoint, 738
distributive laws, 739
empty, 737
finite, 737
function, 737
intersection, 738
mathematical, 737
partition, 738
union, 738
set derivatives, 722
set difference, 738
set element, 737
set integral, 447
settling time, 32
sextant, 634
Shannons sampling theorem, 171
short-period mode, 269
shortest-path problem
 Euclidean, 143
shot noise, 600
side force
 fixed-wing aircraft, 242
sideslip angle, 237
sidewash angle, 759
 equation, 771
sight reduction, 634
sigma plot, 100
sigma-point Kalman filters, 556
sigma-point particle filter, 571
sigma-point Rauch-Tung-Striebel smoothers, 556
sigma-points, 483, 553
signal, 12, 488
 analog, 12
 digital, 12
 discrete-time, 12
 periodic, 39
signal-to-noise ratio, 601
signals of opportunity, 634
significance level, 441
simulation
 system, 10
simultaneous localization and mapping, 677
single-degree LTI feedback control system, 159
single-difference pseudorange
 between-receiver, 648
 between-transmitter, 646
single-differenced phase-based range equation, 652
single-spin stabilized, 283
singular value, 627, 744
 maximum, 745
 minimum, 745
singular value decomposition, 744
singular value plot, 100
SIR step
 BPF, 567
 EKF, 572
 RBPF, 574
 SPPF, 572
SISO feedback control system
 characteristic equation, 63
 characteristic polynomial, 63
 stability, 63
skew-symmetric matrix operator, 667
skewness, 411
Sklar's theorem, 419
slant range, 638
slat, 215
small angle approximation, 23
small gain theorem, 120
small-disturbance theory, 20
small-perturbation theory, 20
Smirnov transform, 419
smoothing step
 Bayes smoother, 492
 extended Rauch-Tung-Striebel smoother, 547
 fixed-interval extended Kalman smoother, 547
 fixed-interval Kalman smoother, 501
 fixed-interval sigma-point Kalman smoother, 558

- fixed-interval statistically linearized Kalman smoother, 552
- fixed-lag Kalman smoother, 510
- Rauch-Tung-Striebel smoother, 501
- sigma-point Rauch-Tung-Striebel smoother, 558
- statistically linearized Rauch-Tung-Striebel smoother, 552
- soft-iron variation, 623
- software-defined radio, 594
- software-in-the-loop, 10
- sojourn time, 432
- sojourn-time, 703
- solution separation, 609
- solution separation method, 611
- Somigliana gravity model, 299
- sonar, 634
- space cone, 279
- space segment, 589
- space vehicles, 589
 - GPS, 593
- spacecraft
 - chaser, 318
 - target, 318
- span
 - wing, 755
- span efficiency
 - wing, 757
- sparsity, 472
- spatial Poisson process, 689
- spawning
 - target, 678
- specific force, 586
- spectral radius, 122
- speed hold, 367
- speed of sound, 756
- spherical-Earth model, 311, 318
- spin angle, 236
- spiral mode, 270
- spoiler, 215
- spoofing, 591
- square matrix, 740
- square-root correction step
- Kalman filter, 506
- square-root Kalman filter, 505
- stabilator, 211
- stability
 - asymptotic, 20
 - global, 20
 - global asymptotic, 20
 - lateral-directional, 270
 - longitudinal, 269
 - LTI MIMO continuous-time, 100
 - LTI system, 28, 173
 - Lyapunov, 20
 - modal continuous-time, 28
 - modal discrete-time, 173
 - SISO feedback control system, 63
- stability and control coefficients
 - longitudinal, 762
- stability and control derivative
 - elastic, 356
- stability and control derivatives, 265
 - lateral-directional, 770
 - longitudinal, 762
- stability augmentation system, 58, 89
- stability augmentation systems
 - airplane, 373
- stability frame, 238, 258
- stability index, 422
- stability margin
 - (time) delay, 66
 - disk, 67
 - gain, 64
 - phase, 65
 - SISO, 64
- stabilizability, 115
- stabilizer
 - horizontal, 210
 - vertical, 210
- stable matrix, 743
- stagnation pressure, 603
- stall, 756
- standard deviation, 411
- standard gravitational acceleration, 241, 711
- standard gravitational parameter, 241

- standard normal distribution, 421
- standard score, 442
- standard time, 617
- standard Wiener process, 433
- standardized moment
 - random variable, 411
- star map, 620, 627
- star tracker, 620, 627
- state
 - base, 435
 - Markov process, 430
 - modal, 435
 - vector, 14, 168
- state controllability, 114
- state controllability matrix
 - discrete-time, 172, 179
- state equation, 14, 434
 - discrete-time, 168
- state error
 - continuous-time, 113, 178
- state estimate, 112, 177
- state estimation, 488
 - vehicle (self), 365
- state extraction step
 - BPF, 568
 - EKPF, 573
 - RBPF, 575
 - SPPF, 573
- state feedback control, 111, 177
- state filter, 488
- state filtering, 488
- state filtering PDF
 - posterior, 492
 - prior, 491, 492
- state filtering posterior PDF, 490, 492
- state matrix
 - continuous-time, 14
 - discrete-time, 168
- state observability
 - discrete-time, 179
- state posterior PDF
 - marginal, 493
- state prediction PDF
 - posterior, 491
- state prediction posterior PDF, 490, 491
- state predictor, 488
- state prior PDF, 489
- state reachability, 114
- state smoother, 488
- state smoothing, 488
- state smoothing PDF
 - posterior, 492
- state smoothing posterior PDF, 490, 492
- state space
 - probability, 406
 - random process, 427
- state transition
 - Markov process, 430
- state transition matrix
 - discrete-state Markov process, 431
- state transition PDF, 489
- state transition probabilities, 431
- state transition probability
 - one-step, 431
- state vector
 - modal, 27, 172
- state-of-health, 606
- state-space
 - continuous-time, 13
 - continuous-time linear, 14
 - continuous-time LTI, 15
 - continuous-time, time-invariant, 15
 - discrete-time, 168
 - discrete-time linear, 168
 - discrete-time LTI, 169
 - discrete-time, time-invariant, 168
- state-space model
 - stochastic, 434
- state-transition matrix
 - discrete-time, 172
 - LTI, 25
 - LTV, 130
- static, 647
- static port
 - blocked, 605
- static pressure, 603

- static-controller feedback, 176
- static-controller feedback control, 111
- stationarity
 - covariance, 429
 - strict-sense, 429
 - wide-sense, 429, 430
- stationary distribution
 - Markov chain, 431
- statistic, 410
 - KS, 408
- statistical inference, 441
- statistical linear regression, 553
- statistical linearization, 549
- statistical regularity, 402
- statistically linearized (Kalman) filter, 550
- statistically linearized Kalman smoother
 - fixed-interval, 551
- statistically linearized Rauch-Tung-Striebel smoother, 551
- steady wind, 305
- steady-flight condition
 - point-mass, 245
- steady-flight conditions, 290
 - rigid-body, 257
- steady-flight equations
 - lateral-directional straight, 259
 - longitudinal straight, 259
 - point-mass alternative, 245
 - rigid-body, 257
 - rigid-body stability frame, 290
- steady-state
 - condition, 32
 - gain, 32
 - input, 32
 - output, 32
- steady-state Kalman filter, 522
- steady-state sinusoidal response, 40
- steepest gradient, 468
- step amplitude, 31
- step response, 31
 - unit, 31
- stereo-vision camera, 602
- Stirling's formula
- second-order , 559
- stochastic decoupling, 483
- stochastic dynamical system, 434
- stochastic OCP
 - fundamental, 436
- stochastic process, 427
- stochastic state-space
 - fault model, 609
- stochastic state-space model
 - discretized LTI, 520
 - linear, 434
- stochastic universal sampler, 567
- straight flight, 246
- straight-line approximation, 43
- strapdown gyroscope, 584, 623, 624
- strapdown sensor, 663
- strict-sense stationarity, 429
- structured singular value, 121
- subset, 737
 - proper, 737
- sum of squares
 - explained, 464
 - regression, 464
 - residual, 463
 - total, 463
- superpositional sensors, 679
- supplementary equation
 - geocentric rates, 312
 - geodetic rates, 313
- support
 - probability, 405
- supremum, 742
- survival probability, 720
- swashplate, 216
- sweep angle, 757
- symmetric congruence transformation, 158
- symmetric matrix, 740
- symmetric wing, 756
- synthetic aperture radar, 602
- system
 - autonomous, 13
 - deterministic, 13
 - dynamical, 12

- general, 12
- linear, 12
- linear, parameter-varying, 131, 164
- mimo, 13
- miso, 13
- mode, 26
- nonlinear, 12
- response, 12
- simo, 13
- siso, 13
- static, 12
- stochastic, 13
- time-invariant, 12
- time-varying, 13
- system faults, 603
- system identification
 - aircraft, 242
 - airplane, 265
- system response
 - free, 31
 - frequency, 41
 - impulse, 34
 - initial condition, 31
 - pole effects, 37
 - step, 31
 - unit step, 31
 - zero effects, 38
 - zero-input, 31
- system type, 123
- T-tail, 212
- tail
 - A-, 214
 - conventional, 210
 - cruciform, 212
 - H-, 212
 - horizontal, 210
 - inverted V-, 213
 - T-, 212
 - twin, 212
 - V-, 213
 - vertical, 210
 - X-, 213
- Y-, 214
- tail distribution, 408
- tail index, 422
- tailerons, 211
- tailless, 210
- Tait-Bryan Euler angles, 228
- tandem wings, 210
- target
 - birth, 678
 - death, 678
 - spawning, 678
- target function, 483
- target motion uncertainty, 679
- target state, 678
- target tracking, 678
 - ballistic, 710
 - compatible tracks, 694
 - extended, 679
 - group, 679
 - point, 679
- taxicab norm, 741
- Taylor series
 - multivariate, 21
 - univariate, 20
- temperative lapse rate, 304
- terminal cost, 134
- ternary-uniform mixture distribution, 702
- terrain map, 603
- terrain-referenced navigation, 677
- terrain-relative navigation, 677
- test statistic, 441
- thermal noise, 600
- third-order spherical cubature rule (TOSCR), 560
- three-axis magnetometer, 623
- three-body problem, 318
- thrust, 241
- thrust coefficient, 769
- thrust vectoring, 241
- thruster
 - pair, 362
 - single, 362
- tight GNSS/INS integration, 672
- tilt compensation, 623

- time
 civil, 617
 coordinated universal, 617
 daylight saving, 617
 posix, 617
 standard, 617
 unix, 617
- time constant
 maneuver, 701
 modal, 28
- time difference of arrival, 635
- time horizon
 continuous-time, 133
 discrete-time, 181
- time of arrival, 619, 635
- time of birth, 733
- time of flight, 634
- time of transmission, 618
- time sample, 427
- time shift operator, 171
- time step, 168
- time synchronization
 GNSS/INS, 677
- time update step
 error-state, 578
- time-homogeneous, 431
- time-invariant
 continuous-time state-space, 15
 discrete-time state-space, 168
 ODE, 14
- time-of-flight equation, 713
- time-varying dynamical system, 126
- timebase, 619
- timepieces, 615
- torque-free motion, 278
- torquer, 361
- total electron count, 660
- total pressure, 603
- total sampling time, 171
- total sum of squares, 463
- track score, 697
- track-oriented multi-hypothesis tracking, 695
- track-to-measurement association, 681
- track-to-sensor data association, 681
- tracking loop aiding
 GNSS, 673
- tracking-domain integration, 673
- transfer function, 16
 discrete-time, 170
 fundamental, 61
 matrix, 18
 pole, 17
 standard form, 17
 zero, 17
- transfer function matrix
 standard, 18
- transient response, 40
- transition probabilities, 431
- transition rate
 Markov jump process, 432
- transmitter-receiver-time triple-difference, 656
- transport rate, 669
- transpose, 740
 matrix, 740
- trial step vector, 469
- triangle inequality, 741
- tricopter, 217
- trilateration, 640
- trim point, 19
- trim state, 19
- triple-difference
 cycle count range equation, 656
- tropospheric delay
 code phase, 657
- true anomaly, 247
- true-state vector, 577
- trust-region, 469
- trust-region methods, 469
- turn compensation, 378
- turn-rate, 705
- turning motion, 704
- twin tail, 212
- two-body approximation, 246
- Type II error, 441, 442
- ultra-rapid orbital errors, 659

- unbiased, 456
- uncertainty
 - measurement origin, 678, 679
 - target motion, 679
- uncertainty set, 117
- uncorrelation, 417
 - random process, 430
- uniform asymptotic stability, 128
- uniform distribution, 419
- uniform stability, 128
- union
 - set, 738
- unitary matrix, 740
- universal time, 616
- unix time, 617
- unmatched uncertainties, 196
- unscented transform, 559
 - higher-order, 559
- unsteady winds, 306
- unstructured uncertainty set, 119
- upper triangular matrix, 740
- user equivalent range error, 657
- user segment, 589
- v-n plot, 295
- V-tail, 213
- validation and verification
 - control system, 10
- value function, 182
 - continuous-time, 135
 - forward, 189
- variable-structure multi-model, 526
- variance, 411
 - M -sample, 617
 - 2-sample, 617
 - Allan, 617
 - estimator, 456
- variance-covariance, 417
- variation
 - cost functional, 134
- variational Bayes, 482
- VB parameter estimator, 482
- vector
 - element, 739
 - norm, 740
 - notation, 739
 - row, 740
- vector auto-regressive process, 435
- vector space, 739
 - dimension, 740
- vehicle, 209
 - flight, 209
- vehicle dynamics and control, 209
- vehicle-centered, inertial frame, 233
- velocimeter, 662
- velocity-turn-climb reference frame, 715
- vertical channel instability, 671
- vertical tail
 - efficiency, 771
 - volume ratio, 771
- virtual control input
 - LTI, 196
 - nonlinear, 199
- virtual reference station, 658
- virtual work, 332
- visual camera, 602
- volume ratio
 - horizontal tail, 763
 - vertical tail, 771
- von Kármán gust model, 306
- Wahba's problem, 626
- waterbed effect, 91
- waypoint, 366
- weak white noise, 429
- weight, 241
- weight matrix, 136, 741
 - cross-, 136
 - endpoint, 136
 - input, 136
 - state, 136
 - terminal, 136
- weighted least-squares problem, 461
- wet delay, 660
- WGS parameters, 592
- wide-lane combination, 658

- wide-sense stationarity, 429, 430
Wiener filter, 522
Wiener process, 432
Wilks test, 444
wind frame, 235
wind shear, 306
wind speed, 236
wind-up
 phase, 658
wing span, 755
wings, 210
wings-level flight, 260
wobble angle, 280
X-tail, 213
Y-tail, 214
yaw angle, 237
yaw damper, 271
Z-test, 442
zero
 left half plane, 38
 right half plane, 38
zero dynamics
 LTI, 197
 nonlinear, 199
zero matrix, 15
zero-effort-miss, 397
zero-input response, 31
zero-lift angle of attack
 wing, 756
Ziegler-Nichols PID tuning method, 89